

ΑΕΠΠ Γ' ΛΥΚΕΙΟΥ

ΣΠΟΥΔΕΣ ΟΙΚΟΝΟΜΙΑΣ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΚΔΟΣΕΙΣ



νέο φροντιστήριο

Τίτλος: ΠΛΗΡΟΦΟΡΙΚΗ

Συγγραφέας: ΔΙΑΛΕΚΤΑΚΗΣ ΓΕΡΑΣΙΜΟΣ

Επιμέλεια
Συντονισμός : ΔΙΑΛΕΚΤΑΚΗΣ ΓΕΡΑΣΙΜΟΣ

Πίνακας περιεχομένων

ΠΡΟΛΟΓΟΣ.....	6
ΚΕΦΑΛΑΙΟ 1	8
1.1 Η έννοια πρόβλημα	8
1.2 Κατανόηση προβλήματος	8
1.3 Ανάλυση προβλήματος	8
1.4 Επίλυση προβλήματος.....	8
1.5 Κατηγορίες προβλημάτων	9
1.6 Πρόβλημα και υπολογιστές	10
ΚΕΦΑΛΑΙΟ 2.....	14
2.1 Τι είναι αλγόριθμος.....	14
2.2 Σπουδαιότητα αλγορίθμων (ΒΛΕΠΕ ΠΑΡΑΛΛΗΛΑ ΚΑΙ ΤΗΝ ΠΑΡΑΓΡΑΦΟ 3.1 ΚΕΦΑΛΑΙΟΥ 3 ΤΕΥΧΟΥΣ Β)	15
Σύμβολα διαγράμματος ροής.....	16
2.4 Βασικές συνιστώσες/εντολές ενός αλγόριθμου.....	17
Βασικά στοιχεία αλγορίθμου	17
Πως γράφουμε ένα ΑΛΓΟΡΙΘΜΟ ή ένα ΠΡΟΓΡΑΜΜΑ	21
2.4.1 Δομή ακολουθίας	22
2.4.2 Δομή Σύνθετης επιλογής	24
2.4.3 Δομές πολλαπλής επιλογής.....	26
2.4.4 Εμφωλευμένες Δομές.....	27
2.4.5 Δομή Επανάληψης.....	29
2.4.5.1 Η Δομή επανάληψης ΓΙΑ.....	29
2.4.5.2 Η Δομή επανάληψης Όσο...Επανάλαβε	31
2.4.5.3 Η Δομή επανάληψης Αρχή_Επανάληψης...Μέχρις_Ότου	33
ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ ΑΠΟ ΟΣΟ σε ΜΕΧΡΙΣ_ΟΤΟΥ	38
ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ ΑΠΟ ΜΕΧΡΙΣ_ΟΤΟΥ σε ΟΣΟ	38
ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ ΣΥΝΘΗΚΩΝ ΑΠΟ ΟΣΟ σε ΜΕΧΡΙΣ_ΟΤΟΥ και ΜΕΧΡΙΣ_ΟΤΟΥ σε ΟΣΟ	39
ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΤΑΣΚΕΥΗΣ ΠΙΝΑΚΑ ΤΙΜΩΝ (Χωρίς Αρίθμηση γραμμών)	41
ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΤΑΣΚΕΥΗΣ ΠΙΝΑΚΑ ΤΙΜΩΝ (Με Αρίθμηση γραμμών).....	42
ΜΕΘΟΔΟΛΟΓΙΑ ΜΕΤΑΤΡΟΠΗΣ ΔΙΑΓΡΑΜΑΤΩΝ ΡΟΗΣ ΣΕ ΑΛΓΟΡΙΘΜΟ / ΨΕΥΔΟΚΩΔΙΚΑ	43
2.5 ΟΛΙΣΘΗΣΗ ΔΕΞΙΑ – ΑΡΙΣΤΕΡΑ	46
2.6 ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΑΛΛΑ ΡΩΣΙΚΑ.....	47
ΜΕΘΟΔΟΛΟΓΙΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕ ΚΛΙΜΑΚΩΤΟ ΥΠΟΛΟΓΙΣΜΟ.....	50
ΕΡΩΤΗΣΕΙΣ ΣΩΣΤΟΥ – ΛΑΘΟΥΣ	53
Ερωτήσεις ΑΠΛΗΣ ΕΠΙΛΟΓΗΣ	56
ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΣΤΟΙΧΗΣΗΣ	59

ΕΡΩΤΗΣΕΙΣ ΣΥΜΠΛΗΡΩΣΗΣ ΚΕΝΩΝ	59
EXTRA ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ	79
ΑΛΥΤΕΣ ΟΜΑΔΕΣ ΑΣΚΗΣΕΩΝ ΕΠΙ ΤΩΝ ΠΙΝΑΚΩΝ ΤΙΜΩΝ - ΛΟΓΙΚΩΝ ΕΚΦΡΑΣΕΩΝ – ΛΟΓΙΚΩΝ ΔΙΑΓΡΑΜΜΑΤΩΝ	83
ΠΙΝΑΚΕΣ ΤΙΜΩΝ	83
ΑΣΚΗΣΕΙΣ ΜΕ ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ	96
ΑΣΚΗΣΕΙΣ ΜΕ ΛΟΓΙΚΑ ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΙ ΜΕΤΑΤΡΟΠΕΣ	98
EXTRA ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ	103
ΕΠΙ ΠΙΝΑΚΩΝ ΤΙΜΩΝ ΚΑΙ ΤΩΝ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ	103
ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ ΑΛΓΟΡΙΘΜΩΝ ΟΜΑΔΑ Α'	108
ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ ΑΛΓΟΡΙΘΜΩΝ ΟΜΑΔΑ Β'	110
ΑΛΥΤΕΣ ΟΜΑΔΕΣ ΑΣΚΗΣΕΩΝ ΣΕ ΑΛΓΟΡΙΘΜΟ	111
ΚΕΦΑΛΑΙΟ 6.....	113
6.1 Η έννοια του προγραμματισμού	113
6.2 Ιστορική αναδρομή.....	114
6.2.1 Γλώσσες μηχανής.....	114
6.2.2 Γλώσσες χαμηλού επιπέδου ή Συμβολικές γλώσσες	114
6.2.3 Γλώσσες υψηλού επιπέδου.....	115
Κατηγοριοποίηση γλωσσών υψηλού επιπέδου:.....	116
6.2.4 Γλώσσες 4ης γενιάς ή ερωταπαντήσεων	118
Οπτικός προγραμματισμός	119
Καθοδηγούμενος από γεγονότα προγραμματισμός.....	119
Ποιά είναι η καλύτερη γλώσσα προγραμματισμού;	119
6.3 Φυσικές και τεχνητές γλώσσες.....	119
6.4 Τεχνικές σχεδίασης προγραμμάτων	121
6.4.1 Ιεραρχική σχεδίαση ή Top down σχεδίαση.....	121
6.4.2 Τμηματικός προγραμματισμός.....	122
6.4.3 Δομημένος προγραμματισμός	122
Πλεονεκτήματα του δομημένου προγραμματισμού.....	122
6.5 Αντικειμενοστραφής προγραμματισμός	123
6.6 Παράλληλος προγραμματισμός.....	126
6.7 Προγραμματιστικά περιβάλλοντα.....	127
GOTO 129	
ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ GOTO ΣΕ ΔΟΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ	130
ΛΥΜΕΝΕΣ ΑΣΚΗΣΕΙΣ ΣΤΗΝ GOTO	133
ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ ΣΤΗΝ GOTO	138
ΕΡΩΤΗΣΕΙΣ ΣΗΜΠΛΗΡΩΣΗΣ ΚΕΝΩΝ	144
ΕΡΩΤΗΣΕΙΣ ΑΠΛΗΣ ΕΠΙΛΟΓΗΣ	145
ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΣΤΟΙΧΗΣΗΣ.....	146

ΚΕΦΑΛΑΙΟ 7	147
7.1 Το αλφάβητο της ΓΛΩΣΣΑΣ	147
7.2 Τύποι δεδομένων	147
7.3 Σταθερές	147
7.4 Μεταβλητές	148
7.5 Αριθμητικοί τελεστές	148
7.6 Συναρτήσεις	149
ΚΕΦΑΛΑΙΟ 8	151
8.1 Εντολές Επιλογής	151
Α. Λογική Έκφραση	151
Β. Σύνθετες Εκφράσεις	151
8.1.1 Εντολή ΑΝ	152
8.2 Εντολές επανάληψης	154
8.2.1 Εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ	154
8.2.2 Εντολή ΜΕΧΡΙΣ_ΟΤΟΥ	154
8.2.3 Εντολή ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ	155
ΓΕΝΙΚΕΣ ΕΡΩΤΗΣΕΙΣ ΣΤΑ ΚΕΦΑΛΑΙΑ 7 - 8	156

ΠΡΟΛΟΓΟΣ

Η έκδοση αυτή έχει ως στόχο να αποτελέσει χρήσιμο εργαλείο για κάθε μαθητή και μαθήτρια αλλά και να παρέχει διευκόλυνση στο έργο των συναδέλφων στην τάξη.

Περιλαμβάνει γνωστικό υλικό για την κατανόηση – ανάπτυξη και μεθοδολογία – ενδεικτικές λύσεις ασκήσεων – ομαδοποιημένες ασκήσεις για τα θέματα του συγκεκριμένου μαθήματος (ΠΛΗΡΟΦΟΡΙΚΗ).

Πάντως το σύγγραμμά αυτό είναι αποτέλεσμα πολλών ετών δουλειάς και έχει όλα τα χαρακτηριστικά εκείνα ώστε να καλύπτουν τους μαθητές στις ιδιαιτερότητες του μαθήματος

Με εκτίμηση

Αθήνα , Μάϊος 2021

ΠΛΗΡΟΦΟΡΙΚΗ
Γ' Λυκείου
ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΣ:
ΣΠΟΥΔΩΝ ΟΙΚΟΝΟΜΙΑΣ & ΠΛΗΡΟΦΟΡΙΚΗΣ

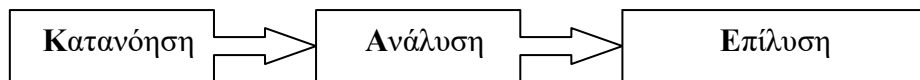
ΑΘΗΝΑ 2021

ΚΕΦΑΛΑΙΟ 1

1.1 Η έννοια πρόβλημα

Πρόβλημα : Εννοούμε μία κατάσταση που χρήζει αντιμετώπισης (επίλυσης), η δε λύση της δεν είναι γνωστή και ούτε προφανής.

Στάδια αντιμετώπισης ενός προβλήματος : (Κ. Α. Ε)



1.2 Κατανόηση προβλήματος

Για την αντιμετώπιση κάθε προβλήματος πρέπει προηγουμένως να έχει προηγηθεί η **κατανόησή** του.

Αποτελεί συνάρτηση δυο παραγόντων:

1. Σωστή διατύπωση εκ μέρους του δημιουργού του,
2. Σωστή ερμηνεία από αυτόν που θα το επιλύσει.

Ορισμοί:

Με τον όρο **δεδομένο** δηλώνεται οποιοδήποτε στοιχείο μπορεί να γίνει αντιληπτό από έναν τουλάχιστον παρατηρητή με μία από τις πέντε αισθήσεις του.

Ο όρος **επεξεργασία δεδομένων** δηλώνει εκείνη τη διαδικασία κατά την οποία ένας “μηχανισμός” δέχεται δεδομένα, τα επεξεργάζεται σύμφωνα με έναν προκαθορισμένο τρόπο και αποδίδει πληροφορίες.

Με τον όρο **πληροφορία** αναφέρεται οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.

1.3 Ανάλυση προβλήματος

Σημαίνει ότι ξεκινάμε να αποκαλύπτουμε τη **δομή του προβλήματος**, δηλαδή να χωρίσουμε το πρόβλημα σε μικρότερα και απλούστερα υπο-προβλήματα, καθένα από τα οποία λύνεται ευκολότερα.

Δομή προβλήματος: Εννοούμε τα συστατικά μέρη από τα οποία συντίθεται, δηλαδή τα επιμέρους τμήματα που το αποτελούν καθώς και τον τρόπο με τον οποίο αυτά συνδέονται μεταξύ τους.

Η ανάλυση μπορεί να γίνει με δομημένο λεκτικό τρόπο ή καλύτερα χρησιμοποιώντας ένα **ιεραρχικό διάγραμμα**.

1.4 Επίλυση προβλήματος

Λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων. Για τη σωστή επίλυση του προβλήματος βασική προϋπόθεση είναι ο καθορισμός απαιτήσεων.

Καθορισμός απαιτήσεων : Σημαίνει να α) προσδιορίσουμε τα **δεδομένα** που παρέχονται β) να προσδιορίσουμε τα **ζητούμενα**, δηλαδή τι περιμένουμε σαν αποτέλεσμα.



- Τα δεδομένα είναι οτιδήποτε στοιχείο μπορεί να γίνει αντιληπτό από τον άνθρωπο.
- Η πληροφορία είναι γνωστικό στοιχείο που προκύπτει από την επεξεργασία δεδομένων.
- Η επεξεργασία δεδομένων είναι μία διαδικασία κατά την οποία ένας μηχανισμός (π.χ. ο Η/Υ) κάνει πράξεις και υπολογισμούς στα δεδομένα ώστε να προκύψει μία χρήσιμη πληροφορία.

Για παράδειγμα, η μέση θερμοκρασία κατά το μήνα Ιανουάριο (πληροφορία) προκύπτει από τον υπολογισμό του μέσου όρου (επεξεργασία) των ημερήσιων θερμοκρασιών του μήνα (δεδομένα).

1.5 Κατηγορίες προβλημάτων

A) Με κριτήριο τη δυνατότητα **επίλυσης** τα διακρίνουμε σε :

Επιλύσιμα	Η λύση τους είναι γνωστή κι έχει διατυπωθεί (π.χ το πρόβλημα του υπολογισμού του εμβαδού του κύκλου).
Ανοικτά	Η λύση τους δεν έχει βρεθεί αλλά παράλληλα δεν έχει αποδειχτεί ότι δεν επιδέχονται λύση (π.χ. το πρόβλημα της ενοποίησης των 4 δυνάμεων της φυσικής).
Άλυτα	Έχουμε παραδεχτεί ότι δεν υπάρχει λύση (πχ. Ο τετραγωνισμός του κύκλου).

B) Με κριτήριο το βαθμό **δόμησης** των λύσεών τους τα διακρίνουμε σε:

Δομημένα	Η λύση τους είναι μία αυτοματοποιημένη διαδικασία. (π.χ. η επίλυση της δευτεροβάθμιας εξίσωσης)
Ημιδομημένα	Η λύση τους μπορεί να προέλθει από πλήθος πιθανών λύσεων που εμείς επιλέγουμε (πχ. Το πρόβλημα του πώς θα πάμε εκδρομή σε ένα μέρος. Μπορούμε να διαλέξουμε ένα πλήθος από πιθανούς τρόπους-τραίνο, αεροπλάνο κλπ).
Αδόμητα	Η λύση τους δεν είναι αυτοματοποιημένη, δηλαδή δεν μπορούμε να βρούμε ένα συγκεκριμένο τρόπο λύσης. Για την επίλυση βασιζόμαστε στην ανθρώπινη διαίσθηση και εμπειρία (πχ ο τρόπος οργάνωσης ενός πάρτι).

Γ) Με κριτήριο το είδος **επίλυσης** τα χωρίζουμε σε:

Απόφασης	Η λύση σε αυτά τα προβλήματα είναι του τύπου «Ναι» και «Όχι». (π.χ. ο αριθμός 101 είναι πρώτος;)
Υπολογιστικά	Για την επίλυση απαιτείται η διενέργεια υπολογισμών. (π.χ. για τον αριθμό N να βρεθεί το παραγοντικό του N!).

Βελτιστοποίησης

Η λύση που ζητάμε είναι το βέλτιστο αποτέλεσμα για τα συγκεκριμένα δεδομένα. (π.χ. ποιος είναι ο συντομότερος δρόμος για να πάμε σε ένα μέρος).

1.6 Πρόβλημα και υπολογιστές

Οι λόγοι που αναθέτουμε την επίλυση ενός προβλήματος σε Η/Υ είναι: (**Π.Ο.Τ.Ε**)

- Η Πολυπλοκότητα των υπολογισμών.
- Ο μεγάλος **Όγκος** των δεδομένων.
- Η Ταχύτητα εκτέλεσης των πράξεων.
- Η Επαναληπτικότητα των διαδικασιών.

Ο Η/Υ, στα κυκλώματά του, εκτελεί **3 μόνο** λειτουργίες: (**Π.Σ.Μ**)

- Πρόσθεση (όλες οι άλλες αριθμητικές πράξεις γίνονται μέσω πρόσθεσης!).
- Σύγκριση (που αποτελεί βασική λειτουργία των λογικών πράξεων).
- Μεταφορά των δεδομένων.

Γενικώς η ικανότητα του Η/Υ εκφράζεται σε ποσοτικό επίπεδο ενώ η ικανότητα του ανθρώπου σε ποιοτικό επίπεδο (Φανταστείτε πόσους υπολογισμούς κάνει ο Η/Υ στο σκάκι για να βρει μία καλή κίνηση ενώ ο άνθρωπος το κάνει με πολύ λιγότερους).

ΕΡΩΤΗΣΕΙΣ ΘΕΩΡΙΑΣ**ΑΝΑΠΤΥΞΗ**

1. Δώστε τον ορισμό του προβλήματος.
2. Τι σημαίνει ο όρος "δεδομένο", "πληροφορία", και "επεξεργασία δεδομένων";
3. Τι εννοούμε με τον όρο καθορισμός απαιτήσεων ενός προβλήματος;
4. Τι είναι η δομή ενός προβλήματος;
5. Περιγράψτε τα τρία στάδια αντιμετώπισης ενός προβλήματος. Σύμφωνα με ποιους τρόπους κατηγοριοποιούμε τα προβλήματα;
6. Περιγράψτε τις κατηγορίες προβλημάτων, σύμφωνα με τη δυνατότητα επίλυσής τους και δώστε αντίστοιχα παραδείγματα
7. Περιγράψτε τις κατηγορίες προβλημάτων, σύμφωνα με το βαθμό δόμησής τους και δώστε αντίστοιχα παραδείγματα.
8. Περιγράψτε τις κατηγορίες προβλημάτων, σύμφωνα με το είδος της επίλυσης που επιζητούμε και δώστε αντίστοιχα παραδείγματα.
9. Για ποιους λόγους μπορεί να αναθέσουμε την επίλυση ενός προβλήματος σε υπολογιστή;
10. Γιατί ανατίθεται η επίλυση ενός προβλήματος σε υπολογιστή;
11. Ποιες είναι οι τρεις λειτουργίες που μπορεί να εκτελέσει ένας υπολογιστής;

ΕΡΩΤΗΣΕΙΣ ΣΩΣΤΟΥ –ΛΑΘΟΥΣ

1. Πρόβλημα είναι μια μαθηματική κατάσταση που πρέπει να αντιμετωπίσουμε
2. Αν υποβάλλουμε τα δεδομένα σε επεξεργασία παίρνουμε πληροφορίες
3. Ο υπολογιστής και το πρόβλημα είναι έννοιες που εξαρτώνται άμεσα η μια από την άλλη
4. Για την επίλυση ενός προβλήματος απαιτείται η σωστή διατύπωσή του
5. Ένα πρόβλημα μπορεί να αναλυθεί σε πολλά επιμέρους προβλήματα
6. Ο ανθρώπινος εγκέφαλος είναι ένας μηχανισμός επεξεργασίας δεδομένων
7. Ο ταχύτερος μηχανισμός επεξεργασίας δεδομένων είναι ο υπολογιστής
8. Η κατανόηση ενός προβλήματος ακολουθεί την ανάλυσή του
9. Ο Η/Υ δεν μπορεί να επιτελέσει όλες τις λειτουργίες του ανθρώπινου εγκεφάλου
10. Η χρήση Η/Υ για την επίλυση προβλημάτων ενδείκνυται στις περιπτώσεις που χρειάζεται διαχείριση μεγάλου όγκου δεδομένων
11. Αν ένα πρόβλημα απαιτεί απλούς υπολογισμούς σε μικρό όγκο δεδομένων δεν μπορεί να ανατεθεί σε έναν Η/Υ
12. Ο Η/Υ μπορεί να επιλύσει με άνεση οποιοδήποτε πολύπλοκο πρόβλημα χωρίς τη βοήθεια του ανθρώπου
13. Για κάθε πρόβλημα υπάρχει και μοναδικός αλγόριθμος επίλυσής του
14. Με τη χρήση Η/Υ μπορούμε να επιλύσουμε οποιοδήποτε πρόβλημα
15. Για την επίλυση ενός προβλήματος πρέπει να έχουν καθοριστεί τα δεδομένα και τα ζητούμενα
16. Τα προβλήματα για τα οποία δεν μπορούμε να απαντήσουμε ακόμη, εάν είναι δυνατόν να επιλυθούν ονομάζονται μη επιλύσιμα
17. Η επίλυση της πρωτοβάθμιας εξίσωσης αποτελεί αδόμητο πρόβλημα
18. Άλυτα ονομάζουμε τα προβλήματα των οποίων η λύση δεν έχει βρεθεί
19. Δομή ενός προβλήματος είναι η εύρεση του συνόλου των μερών που το απαρτίζουν
20. Τα δεδομένα ενός προβλήματος είναι πάντοτε κάποιοι αριθμοί
21. Για την παραγωγή πληροφοριών απαιτούνται δεδομένα ή άλλες πληροφορίες
22. Ένα δομημένο πρόβλημα είναι πάντοτε επιλύσιμο
23. Ένα επιλύσιμο πρόβλημα είναι πάντοτε δομημένο

24. Με τη χρήση αλγορίθμων επιλύονται όλα τα προβλήματα
25. Αλγόριθμος είναι μια "συνταγή" που ορίζει τι πρέπει να γίνει ώστε να φτάσουμε στον επιθυμητό σκοπό
26. Ένα άλυτο πρόβλημα είναι και αδόμητο
27. Η πρόσθεση είναι μια από τις βασικές λειτουργίες που μπορεί να επιτελέσει έναν Η/Υ
28. Πληροφορία είναι το αποτέλεσμα από την επεξεργασία των δεδομένων
29. Πριν από την επίλυση ενός προβλήματος πρέπει αυτό να έχει διατυπωθεί με ακρίβεια και σαφήνεια
30. Με την επεξεργασία πληροφοριών μπορούν να εξαχθούν και άλλες πληροφορίες
31. Η κατανόηση ενός προβλήματος εξαρτάται μόνο από την διατύπωσή του
32. Στη δομή ενός προβλήματος περιλαμβάνονται τα συστατικά του μέρη
33. Το ότι το ύψος ενός ατόμου είναι 1,90 αποτελεί δεδομένο, ενώ είναι πληροφορία ότι το άτομο αυτό είναι ψηλό
34. Ανοικτά είναι τα προβλήματα που δεν είναι άλυτα ούτε επιλύσιμα
35. Η κακή διατύπωση ενός προβλήματος μπορεί να οδηγήσει στην μη επίλυσή του
36. Ο υπολογισμός του εμβαδού ενός τριγώνου είναι ανοικτό πρόβλημα
37. Ο υπολογισμός του εμβαδού ενός τριγώνου είναι δομημένο πρόβλημα
38. Με κριτήριο την δυνατότητα επίλυσης ενός προβλήματος οι κατηγορίες είναι: επιλύσιμα, υπολογιστικά και άλυτα

ΕΡΩΤΗΣΕΙΣ ΣΥΜΠΛΗΡΩΣΗΣ ΚΕΝΩΝ

1. Τα δεδομένα μπορούν να παρέχουν πληροφορίες όταν υποβάλλονται σε _____
2. Το πρόβλημα μεγιστοποίησης των κερδών μιας επιχείρησης είναι πρόβλημα _____
3. Για την επίλυση ενός προβλήματος πρέπει να γίνει ο καθορισμός _____
4. Η _____ είναι η βάση της επίλυσης ενός προβλήματος
5. Σημαντικός παράγοντας στην κατανόηση ενός προβλήματος είναι η _____ του
6. Τα συστατικά μέρη που αποτελούν ένα πρόβλημα προσδιορίζουν τη _____ του
7. Τα στάδια αντιμετώπισης προβλήματος είναι:
 1. Κατανόηση => _____ => Επίλυση
8. Με τον όρο _____ προβλήματος αναφερόμαστε στα συστατικά μέρη του προβλήματος

ΕΡΩΤΗΣΕΙΣ ΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

1. Τα δεδομένα ενός προβλήματος πρέπει
 - α) να έχουν καθοριστεί με σαφήνεια
 - β) να είναι δομημένα
 - γ) να είναι αριθμητικά
2. Η διαδικασία μέσω της οποίας βρίσκουμε το ζητούμενο ενός προβλήματος ονομάζεται
 - α) επίλυση
 - β) ανάλυση
 - γ) αξιολόγηση
 - δ) εύρεση
3. Οι λόγοι για τους οποίους αναθέτουμε την επίλυση προβλημάτων σ' έναν Η/Υ είναι:
 - α) Ταχύτητα εκτέλεσης πράξεων
 - β) Χειρισμός μεγάλου όγκου δεδομένων
 - γ) Ικανότητα εκτέλεσης συγκρίσεων
 - δ) Ικανότητα για ανάλυση δεδομένων
 - ε) Μπορεί να λύσει όλα τα υπολογιστικά προβλήματα δεδομένων

4. Οι βασικές λειτουργίες που μπορεί να επιτελέσει ένας Η/Υ είναι:

- α) πολλαπλασιασμός
- β) Χειρισμός μεγάλου όγκου δεδομένων
- γ) μεταφορά δεδομένων
- δ) Ικανότητα για ανάλυση δεδομένων
- ε) σύγκριση
- στ) δυνάμεις

ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΣΤΟΙΧΗΣΗΣ

1. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β (κάθε στοιχείο της στήλης Α μπορεί να ταιριάζει με περισσότερα στοιχεία της στήλης Β)

A Τιμή	B Τύπος Δεδομένων
1. Πρόβλημα υπολογιστικό	A) Εύρεση εμβαδού κύκλου
2. Πρόβλημα βελτιστοποίησης	B) Αγορά αυτοκινήτου
3. Πρόβλημα απόφασης	Γ) Εύρεση γρηγορότερης διανομής γραμμάτων
	Δ) Πόσοι μαθητές θα πάρουν αριστείο
	Ε) Ένας αριθμός είναι άρτιος
	ΣΤ) Ο μαθητής Ιωάννου θα πάρει αριστείο

2. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β

A Τιμή	B Τύπος Δεδομένων
1. Ανοικτό	A) Έχουμε φτάσει στην παραδοχή ότι δεν επιδέχονται λύση
2. Δομημένο	B) Απαντά σε ένα ερώτημα με ένα "Ναι" ή "Όχι"
3. Απόφασης	Γ) Η λύση προέρχεται από μια αυτοματοποιημένη διαδικασία
	Δ) Η λύση τους δεν έχει βρεθεί αλλά δεν έχει αποδειχτεί ότι δεν επιδέχονται λύση
	Ε) Η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων
	ΣΤ) Ζητάμε το βέλτιστο αποτέλεσμα για τα δεδομένα του προβλήματος

ΚΕΦΑΛΑΙΟ 2

2.1 Τι είναι αλγόριθμος

Αλγόριθμος, είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

Τα **πέντε κριτήρια** που πρέπει να ικανοποιεί ένας αλγόριθμος είναι: (**Ε.Ε.Κ.Π.Α**)

1. **Είσοδος (input).**
2. **Έξοδος (output).**
3. **Καθοριστικότητα (definiteness).**
4. **Περατότητα (finiteness).**
5. **Αποτελεσματικότητα (effectiveness).**

Είσοδος	Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών..
Έξοδος	Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.
Καθοριστικότητα	Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Λόγου χάριν, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση όπου ο διαιρέτης λαμβάνει μηδενική τιμή.
Περατότητα	Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολογιστική διαδικασία (computational procedure).
Αποτελεσματικότητα	Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

Η έννοια του αλγορίθμου **ΔΕΝ ΣΥΝΔΕΕΤΑΙ ΑΠΟΚΛΕΙΣΤΙΚΑ ΚΑΙ ΜΟΝΟ** με προβλήματα της πληροφορικής.

Ας θεωρήσουμε ότι θέλουμε να γευματίσουμε και επομένως πρέπει να εκτελέσουμε τις επόμενες ενέργειες: να συγκεντρώσουμε τα υλικά, να προετοιμάσουμε τα σκεύη μαγειρικής, να παρασκευάσουμε το φαγητό, να ετοιμάσουμε τη σαλάτα, να στρώσουμε το τραπέζι, να γευματίσουμε, να καθαρίσουμε το τραπέζι, και να πλύνουμε τα πιάτα και τα κουζινικά. Είναι ευνόητο ότι η προηγούμενη αλληλουχία των ενεργειών οδηγεί στο επιθυμητό αποτέλεσμα. Βέβαια, αυτή η αλληλουχία δεν είναι η μοναδική για την επίτευξη του σκοπού, αφού, για παράδειγμα, μπορούμε πρώτα να ετοιμάσουμε τη σαλάτα και μετά να

παρασκευάσουμε το φαγητό, ενώ ακόμη μπορούμε πρώτα να πλύνουμε τα πιάτα και μετά να καθαρίσουμε το τραπέζι.

Ωστόσο, το παράδειγμα θέλει να δείξει ότι η θεώρηση μίας σύνθετης εργασίας με διακριτά βήματα που εκτελούνται διαδοχικά είναι ένας πολύ χρήσιμος και πρακτικός τρόπος σκέψης για την επίλυση πολλών (αν όχι όλων των) προβλημάτων.

2.2 Σπουδαιότητα αλγορίθμων (ΒΛΕΠΕ ΠΑΡΑΛΛΗΛΑ ΚΑΙ ΤΗΝ ΠΑΡΑΓΡΑΦΟ 3.1 ΚΕΦΑΛΑΙΟΥ 3 ΤΕΥΧΟΥΣ Β)

Η έννοια του αλγορίθμου είναι θεμελιώδης για την επιστήμη της Πληροφορικής. Η μελέτη των αλγορίθμων είναι πολύ ενδιαφέρουσα, γιατί είναι η πρώτη ύλη για τη μελέτη και εμβάθυνση, αν όχι σε όλες, τουλάχιστον σε πάρα πολλές γνωστικές περιοχές της επιστήμης αυτής.

Η Πληροφορική, λοιπόν, μπορεί να ορισθεί ως η επιστήμη που μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές:

- **Υλικού** (hardware). Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία αρχιτεκτονική τα διάφορα συστατικά του υπολογιστή (δηλαδή ανάλογα με το αν ο υπολογιστής έχει κρυφή μνήμη και πόση, ανάλογα με την ταχύτητα της κύριας και δευτερεύουσας μνήμης κ.ο.κ.).
- **Γλωσσών Προγραμματισμού** (programming languages). Το είδος της γλώσσας προγραμματισμού που χρησιμοποιείται (**δηλαδή, χαμηλότερου ή υψηλότερου επιπέδου**) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου. Γενικά μία γλώσσα που είναι χαμηλότερου επιπέδου (όπως η **assembly** ή η γλώσσα **C**) είναι ταχύτερη από μία άλλη γλώσσα που είναι υψηλότερου επιπέδου (όπως η **Basic** ή **Pascal**). Ακόμη, σημειώνεται ότι διαφορές συναντώνται μεταξύ των γλωσσών σε σχέση με το πότε εμφανίσθηκαν. Για παράδειγμα, παλαιότερα μερικές γλώσσες προγραμματισμού δεν υποστήριζαν την αναδρομή (έννοια που θα εξετάσουμε σε βάθος αργότερα).
- **Θεωρητική** (theoretical). Το ερώτημα που συχνά τίθεται είναι αν πράγματι υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος. Η εξέταση αυτού του ερωτήματος είναι δύσκολο να σχολιασθεί στα πλαίσια του βιβλίου αυτού, επειδή απαιτεί μεγάλη θεωρητική κατάρτιση. Ωστόσο η προσέγγιση αυτή είναι ιδιαίτερα σημαντική, γιατί προσδιορίζει τα όρια της λύσης που θα βρεθεί σε σχέση με ένα συγκεκριμένο πρόβλημα.
- **Αναλυτική** (analytical). Μελετώνται οι υπολογιστικοί πόροι (**computerresources**) που απαιτούνται από έναν αλγόριθμο, όπως για παράδειγμα το μέγεθος της κύριας και της δευτερεύουσας μνήμης, ο χρόνος για λειτουργίες **CPU** και για λειτουργίες εισόδου/εξόδου κ.λπ. Το αντικείμενο αυτό θα εξηγηθεί πληρέστερα στο Κεφάλαιο 5.

2.3 Περιγραφή και αναπαράσταση αλγορίθμων

Οι **τρόποι αναπαράστασης** ενός αλγορίθμου είναι:

1) **Με ελεύθερο κείμενο:** Αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί εύκολα να οδηγήσει σε **μη εκτελέσιμη παρουσίαση** παραβιάζοντας το τελευταίο χαρακτηριστικό των αλγορίθμων, δηλαδή **την αποτελεσματικότητα**.

2) **Με διαγραμματικές τεχνικές:**

Η πιο γνωστή τεχνική είναι το **διάγραμμα ροής** (flowchart) που παλαιότερα ήταν αρκετά δημοφιλής, που συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο παλιά και η πιο γνωστή ίσως, είναι το διάγραμμα ροής (flow chart). Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτελεί την καλύτερη λύση, γι' αυτό και εμφανίζονται όλο και σπανιότερα στη βιβλιογραφία και στην πράξη

3) **Με φυσική γλώσσα κατά βήματα:**

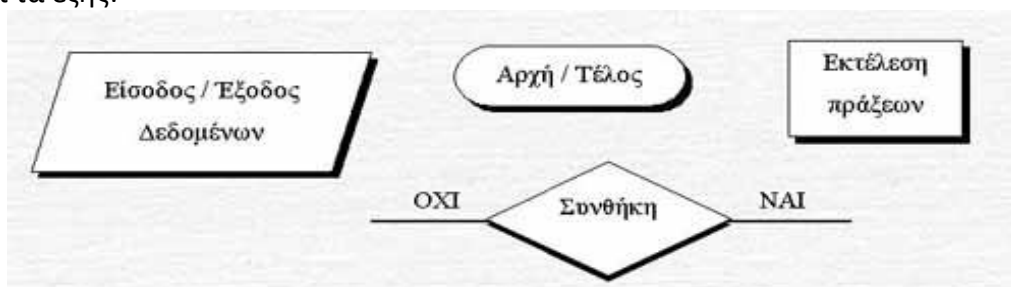
Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί το τρίτο βασικό χαρακτηριστικό ενός αλγορίθμου, όπως προσδιορίσθηκε προηγουμένως, δηλαδή το κριτήριο **της καθοριστικότητας**. Δεν είναι προτιμητέος τρόπος.

4) **Με κωδικοποίηση:**

Δηλαδή με ένα πρόγραμμα γραμμένο είτε σε μία ψευδογλώσσα είτε σε κάποια γλώσσα προγραμματισμού που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο. Στην περίπτωση αυτή, δημιουργούμε τις εντολές κατευθείαν σε μία γλώσσα προγραμματισμού.

Σύμβολα διαγράμματος ροής

Ένα διάγραμμα ροής αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια ή λειτουργία. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών. Τα κυριότερα χρησιμοποιούμενα γεωμετρικά σχήματα είναι τα εξής:



Ελλειψη

Δηλώνει την αρχή και το τέλος του κάθε αλγορίθμου.

Ρόμβος

Δηλώνει μία ερώτηση με δύο ή περισσότερες εξόδους για απάντηση.

Ορθογώνιο παραλληλόγραμμο

Για απάντηση που δηλώνει την εκτέλεση μίας ή περισσότερων πράξεων.

Πλάγιο παράλληλογράμμο

Δηλώνει είσοδο ή έξοδο στοιχείων. Πολλές φορές το σχήμα αυτό μπορεί να διαφοροποιείται προκειμένου να προσδιορίζεται και το είδος της συσκευής απ' όπου γίνεται η είσοδος ή η έξοδος(Πληκτρολόγιο, οθόνη, εκτυπωτή)

2.4 Βασικές συνιστώσες/εντολές ενός αλγόριθμου

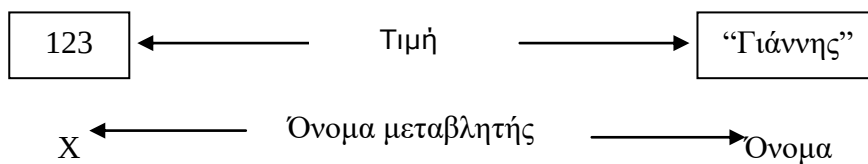
Βασικά στοιχεία αλγορίθμου

1) **Σταθερές:** Με τον όρο αυτό αναφερόμαστε σε **προκαθορισμένες** τιμές που παραμένουν αμετάβλητες σε όλη τη διάρκεια της εκτέλεσης ενός αλγορίθμου. Δηλαδή αφορούν ποσότητες που δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Υπάρχουν 3 ειδών :

- **Αριθμητικές** π.χ. 123, -8, 3.14
- **Αλφαριθμητικές (χαρακτήρες)** π.χ. "Γιάννης", "Μακεδονίας 12". Λέγονται και λεκτικά ή συμβολοσειρές. Περικλείονται σε " "
- **Λογικές** π.χ. ΑΛΗΘΗΣ, ΨΕΥΔΗΣ.

2) **Μεταβλητές:** Είναι ένα γλωσσικό αντικείμενο, που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου. Στη μεταβλητή εκχωρείται μια τιμή, η οποία **μπορεί να αλλάζει** κατά τη διάρκεια εκτέλεσης του αλγορίθμου.

Ανάλογα με το είδος της τιμής που μπορούν να λάβουν, οι μεταβλητές διακρίνονται σε αριθμητικές, αλφαριθμητικές και λογικές..



Στις μεταβλητές δίνουμε ένα όνομα (αναγνωριστικό) και αυτό χρησιμοποιούμε στον αλγόριθμο.

(Τα ονόματα των μεταβλητών ακολουθούν τον εξής κανόνα : Ελληνικά ή Αγγλικά, κεφαλαία ή μικρά, από σύμβολα μόνο η κάτω παύλα `_`, οι αριθμοί επιτρέπονται αλλά όχι σαν πρώτος χαρακτήρας, και απαγορεύονται ονόματα που είναι δεσμευμένες λέξεις, π.χ απαγορεύεται όνομα μεταβλητής Διάβασε ή αλγόριθμος).

Η τιμή της αλλάζει με μία εντολή εκχώρησης τιμής

π.χ. $X \leftarrow 123$
ή
 $\text{Όνομα} \leftarrow \text{"Γιάννης"}$.

Κι εδώ, ανάλογα με το είδος δεδομένου που εκχωρείται η μεταβλητή διακρίνεται σε: (**X**, **Λ**, **Α**, **Π**)

- **Αριθμητική** π.χ. $X \leftarrow 123$
- **Αλφαριθμητική (χαρακτήρας)** π.χ. $\text{Όνομα} \leftarrow \text{"Γιάννης"}$
- **Λογική** π.χ. $\text{Έγγαμος} \leftarrow \text{Ψευδής}$.

Επιπλέον, οι **αριθμητικές** διακρίνονται σε

Ακέραιες,	αν δέχονται ακέραια τιμή	π.χ.	5 , -12
Πραγματικές,	αν δέχονται πραγματική τιμή	π.χ.	3.14 , -4.56

3) Τελεστές: Είναι τα γνωστά σύμβολα. Είναι 3 ειδών:

- **Αριθμητικοί:** +, -, * (πολλ/σμός), / (διαίρεση), ^ (ύψωση σε δύναμη), DIV (ακέραια διαίρεση), MOD (ακέραιο υπόλοιπο).

Σειρά εκτέλεσης αριθμητικών πράξεων :

1^ο το ^,

2^α και με ίδια προτεραιότητα τα *, /, div, mod και από αριστερά προς τα δεξιά

3^α τα +, - και φυσικά πρώτα οι παρενθέσεις

Π.χ. $X \leftarrow 5 * 2$, $X \leftarrow 6^2 (=36)$, $X \leftarrow 5 \text{ DIV } 2 (=2 \text{ πηλίκο})$, $X \leftarrow 5 \text{ MOD } 2 (=1 \text{ υπόλοιπο})$, $X \leftarrow 4/2 * 2=4$ (εδώ πρώτα η διαίρεση γιατί ξεκινάμε από αριστερά προς τα δεξιά), ενώ $X \leftarrow 2 * 2/4=1$ (εδώ πρώτα ο πολλαπλασιασμός γιατί ξεκινάμε από αριστερά προς τα δεξιά), $X \leftarrow 50 \text{ div } 22 * 11 \text{ mod } 4=2$ (από αριστερά προς τα δεξιά)

- **Λογικοί :** ΚΑΙ, Η, ΟΧΙ.

Οι λογικοί τελεστές χρησιμοποιούνται για την εκτέλεση λογικών πράξεων.

Μια λογική πράξη εκτελείται μεταξύ λογικών παραστάσεων (κάθε μία από τις οποίες έχει τιμή (**ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**)) και έχει τελικά αποτέλεσμα **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**. Οι λογικές πράξεις που θα μας απασχολήσουν είναι:

- Η άρνηση (**ΟΧΙ**)
- Η σύζευξη (**ΚΑΙ**)
- Η διάζευξη (**Η**)

Ο πίνακας τιμών των λογικών πράξεων ΚΑΙ, Ή, ΟΧΙ για δύο τελεστέους **A** και **B** φαίνεται στον ακόλουθο πίνακα:

A	B	A ΚΑΙ B	A Ή B	ΟΧΙ A
ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ
ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ
ΑΛΗΘΗΣ	ΨΕΥΔΗΣ	ΨΕΥΔΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ
ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ

Με τη βοήθεια των τριών λογικών πράξεων μπορούμε να δημιουργήσουμε και πιο σύνθετες παραστάσεις. Στις σύνθετες παραστάσεις η σημασία των παρενθέσεων είναι καθοριστική.

Σειρά εκτέλεσης λογικών πράξεων :

1^ο το όχι και

2^α με την ίδια προτεραιότητα τα ΚΑΙ, Ή από αριστερά προς τα δεξιά και φυσικά πρώτα οι παρενθέσεις

π.χ. Έγκυρος $\leftarrow (X \geq 1) \text{ AND } (X \leq 20)$

- **Συγκριτικοί:** $>$, $\geq$, $<$, $\leq$, $<>$ (διάφορο), $=$

4) Εκφράσεις: Οι εκφράσεις διαμορφώνονται από τους **τελεστέους** (operands), που είναι **σταθερές** και **μεταβλητές** και από τους **τελεστές** που είναι **οι αριθμητικοί - συγκριτικοί και λογικοί**. Η διεργασία αποτίμησης μιας έκφρασης συνίσταται στην απόδοση τιμών στις μεταβλητές και στην εκτέλεση των πράξεων. Η τελική τιμή μιας έκφρασης εξαρτάται από την ιεραρχία των πράξεων και τη χρήση των παρενθέσεων. Μια έκφραση μπορεί να αποτελείται από μια μόνο μεταβλητή ή σταθερά μέχρι μια πολύπλοκη μαθηματική παράσταση.

5) Βασικές εντολές:

- **Εντολή εισόδου:** **Διάβασε λίστα μεταβλητών**
π.χ. Διάβασε X
Στη μεταβλητή X εισάγεται μία τιμή.

Συντάσσεται με την λέξη **Διάβασε** και δεξιά από αυτή το **όνομα μιας μεταβλητής (ή περισσότερων)**, είναι **εντολή εισόδου**, που σημαίνει ότι χρησιμοποιείται για την **είσοδο δεδομένων** από το πληκτρολόγιο.

π.χ. **Διάβασε X** ή **Διάβασε X,Y,Z** (για περισσότερες μεταβλητές χωρίζω με κόμμα)

- **Εντολή εξόδου:**
1. **ΣΕ ΑΛΓΟΡΙΘΜΟ**

I) Εμφάνισε λίστα στοιχείων

Π.χ. **Εμφάνισε X**

Είναι **εντολή εξόδου**, που σημαίνει ότι χρησιμοποιείται για εμφάνιση πληροφοριών στην οθόνη.

π.χ.

Εμφάνισε X

Εμφάνισε X,Y

Εμφάνισε ' Καλημέρα '

Εμφάνισε ' Το εμβαδό του τριγώνου είναι: ', X

II) Εκτύπωσε λίστα στοιχείων

Π.χ. **Εκτύπωσε X**

Είναι **εντολή εξόδου**, που σημαίνει ότι χρησιμοποιείται για εκτύπωση πληροφοριών στον εκτυπωτή.

π.χ.

Εκτύπωσε X

Εκτύπωσε X,Y

Εκτύπωσε ' Καλημέρα '

Εκτύπωσε ' Το εμβαδό του τριγώνου είναι: ', X

2) ΣΕ ΠΡΟΓΡΑΜΜΑ

Γράψε λίστα στοιχείων

Π.χ. ΓΡΑΨΕ Χ

Είναι **εντολή εξόδου**, που σημαίνει ότι χρησιμοποιείται για εμφάνιση πληροφοριών στην οθόνη.

π.χ.

ΓΡΑΨΕ Χ

ΓΡΑΨΕ Χ,Υ

ΓΡΑΨΕ ' Καλημέρα '

ΓΡΑΨΕ ' Το εμβαδό του τριγώνου είναι: ', Χ

- **Εντολή εκχώρησης τιμής:** Μορφή: **Μεταβλητή ← Έκφραση**
Αριστερά βάζουμε το όνομα της μεταβλητής και δεξιά μία τιμή ή έκφραση

Π.χ. **$X \leftarrow (5 * 2) / 100$**

Το αποτέλεσμα εκχωρείται στη μεταβλητή.

Με δύο τρόπους λοιπόν, τοποθετούμε τιμές σε μεταβλητές:

- | | |
|--|----------------------------|
| 1. Με εντολή εισόδου Διάβασε | π.χ. Διάβασε ηλικία |
| 2. Κατευθείαν με μία εντολή εκχώρησης. | π.χ. ηλικία ← 18 |

Η **πρώτη γραμμή** ενός αλγορίθμου ξεκινά με την λέξη **Αλγόριθμος** και δεξιά το **όνομα του αλγορίθμου**, το οποίο όνομα ακολουθεί τον κανόνα των ονομάτων που μάθαμε και για τις μεταβλητές. Η τελευταία γραμμή του αλγορίθμου ξεκινά με την λέξη **Τέλος** και δεξιά το **όνομα του αλγορίθμου** το οποίο βάλαμε και στην αρχή.

Η **πρώτη γραμμή** ενός **ΠΡΟΓΡΑΜΜΑΤΟΣ** ξεκινά με την λέξη **ΠΡΟΓΡΑΜΜΑ** και δεξιά το **όνομα ΠΡΟΓΡΑΜΜΑΤΟΣ**, το οποίο όνομα ακολουθεί τον κανόνα των ονομάτων που μάθαμε και για τις μεταβλητές. Η τελευταία γραμμή του **ΠΡΟΓΡΑΜΜΑΤΟΣ** ξεκινά με την λέξη **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ** και δεξιά προαιρετικά το **όνομα του ΠΡΟΓΡΑΜΜΑΤΟΣ** το οποίο βάλαμε και στην αρχή.

Αλγόριθμος Όνομα_αλγορίθμου

Εντολές

Τέλος Όνομα_αλγορίθμου

ΠΡΟΓΡΑΜΜΑ ΌΝΟΜΑ_ΠΡΟΓΡΑΜΜΑΤΟΣ

!ΤΜΗΜΑ ΔΗΛΩΣΕΩΝ ΣΤΑΘΕΡΩΝ

ΣΤΑΘΕΡΕΣ

$\pi=3.14$

!ΤΜΗΜΑ ΔΗΛΩΣΕΩΝ ΜΕΤΑΒΛΗΤΩΝ

ΜΕΤΑΒΛΗΤΕΣ

!Δηλώνουμε τις μεταβλητές π.χ.

Χαρακτήρες: ΟΝΟΜΑ

Λογικές: ΒΡΕΘΗΚΕ

Ακέραιες : X,Y,Z

Πραγματικές: ΜΕΣΟΣ_ΟΡΟΣ

ΑΡΧΗ

Εντολές

Τέλος_προγράμματος Εμβαδό_τριγώνου

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

2.4.1 Δομή ακολουθίας

Η ακολουθιακή δομή εντολών (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση **απλών προβλημάτων**, όπου είναι **δεδομένη** η σειρά εκτέλεσης ενός συνόλου ενεργειών. Ένα απλό παράδειγμα από την καθημερινή ζωή είναι η ακολουθία οδηγιών μίας συνταγής μαγειρικής με στόχο την κατασκευή ενός φαγητού. Τα βήματα και οι ποσότητες που πρέπει να ακολουθηθούν είναι συγκεκριμένα και οι οδηγίες απόλυτα καθορισμένες και σαφείς.

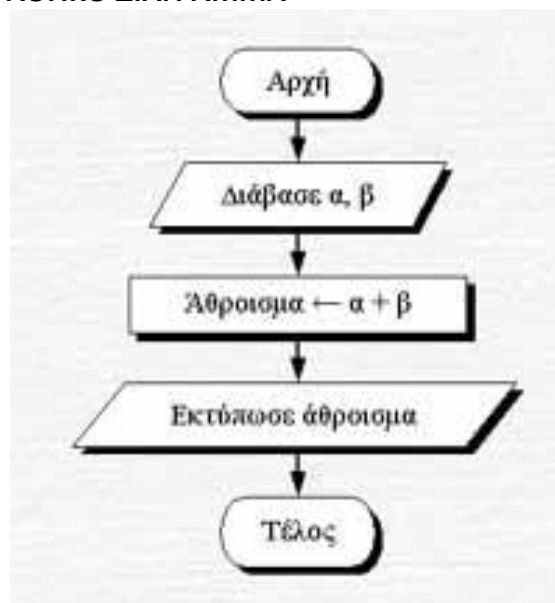
Το παράδειγμα που ακολουθεί παρουσιάζει ένα απλό πρόβλημα που επιλύεται με σειριακή εκτέλεση εντολών.

Παράδειγμα : Ανάγνωση και εκτύπωση αθροίσματος 2 αριθμών.

ΑΛΓΟΡΙΘΜΟΣ

Αλγόριθμος Παράδειγμα
Διάβασε α, β
 $\text{άθροισμα} \leftarrow \alpha + \beta$
Εκτύπωσε άθροισμα
Τέλος Παράδειγμα

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ



Σχολιασμός ψευδοκώδικα:

- Οι εντολές εκτελούνται σειριακά όπως παρατίθενται στον αλγόριθμο.
- Στην αρχή του αλγορίθμου πραγματοποιείται η είσοδος των δεδομένων και τελευταία ενέργεια είναι η έξοδος – εκτύπωσή τους.
- Πάντοτε ξεκινάμε με τη λέξη **Αλγόριθμος** και ακολουθεί το όνομα του αλγορίθμου. Το όνομα ενός αλγορίθμου ακολουθεί τους ίδιους κανόνες με το όνομα μιας μεταβλητής. Επιπλέον, δεν μπορεί να χρησιμοποιείται ίδιο όνομα για μια μεταβλητή και για τον αλγόριθμο. Αυτές οι λέξεις καθώς και όλες όσες είναι έντονα γραμμένες καλούνται **δεσμευμένες** λέξεις και δεν επιτρέπεται να παρουσιαστούν αλλιώς, ούτε να χρησιμοποιηθούν με λάθος τρόπο.
- Για την χρησιμοποίηση της τιμής κάποιας μεταβλητής πρέπει οπωσδήποτε να έχει προηγηθεί εκχώρηση τιμής σε αυτή τη μεταβλητή είτε με εντολή εκχώρησης **Διάβασε** είτε με είσοδο δεδομένων από το χρήστη. Πιο πρακτικά, δεν μπορούμε να χρησιμοποιήσουμε μια μεταβλητή στο δεξί τμήμα μιας εντολής εκχώρησης ή σε μια εκτύπωση αν προηγουμένως δεν έχει λάβει τιμή στον αλγόριθμο – **αρχικοποίηση**.

π.χ. $\alpha \leftarrow 5$
 $\beta \leftarrow 4$

- Η πρώτη ή οι πρώτες εντολές πραγματοποιούν είσοδο δεδομένων. Αυτό μπορεί να γίνει με την εντολή **Διάβασε** ακολουθούμενη από τα ονόματα μεταβλητών ή με την εντολή **Δεδομένα** (δες αλγόριθμο).

Όταν η εκφώνηση αναφέρει **ΡΗΤΩΣ** την ανάγνωση των δεδομένων από το χρήστη τότε χρησιμοποιείται η εντολή **ΔΙΑΒΑΣΕ**.

Σε διαφορετική περίπτωση π.χ. όταν η εκφώνηση αναφέρει «**έστω μεταβλητή N που περιέχει το πλήθος αριθμών**», τότε μπορούμε να εισάγουμε την τιμή της μεταβλητής N στον αλγόριθμό μας με την εντολή **Δεδομένα**.

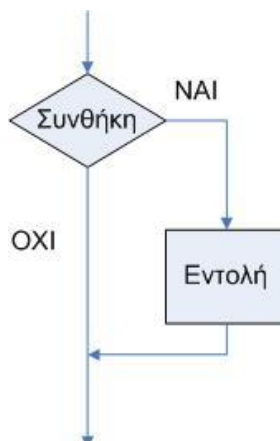
Όμοια για την έξοδο και τις εντολές **Εκτύπωσε** (για αποστολή εκτύπωσης στον εκτυπωτή), **Εμφάνισε** (για εμφάνιση των στοιχείων στην οθόνη) και της εντολής **Αποτελέσματα**.

2.4.2 Δομή επιλογής

Πολλές φορές για να λυθεί ένα πρόβλημα πρέπει να ελεγχθεί αν ισχύει κάποια συνθήκη. Η γενική σύνταξη της δομής επιλογής είναι:

Σύνταξη

Αν συνθήκη **τότε**
 Εντολή
Τέλος_αν



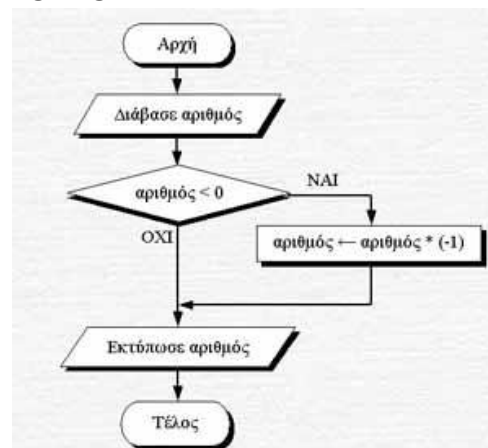
Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει έναν αριθμό και να επιστρέφει την απόλυτη τιμή του. Ο εισαγόμενος αριθμός ελέγχεται αν είναι θετικός ή αρνητικός.

Στην πρώτη περίπτωση δεν χρειάζεται να γίνει καμιά ενέργεια, ενώ στη δεύτερη πρέπει να πολλαπλασιαστεί ο αριθμός με το -1 ώστε να γίνει θετικός.

ΑΛΓΟΡΙΘΜΟΣ

Αλγόριθμος Παράδειγμα
Διάβασε αριθμός
Αν αριθμός < 0 **τότε**
 αριθμός $\leftarrow$ αριθμός $\cdot (-1)$
Τέλος_Αν
Εκτύπωσε αριθμός
Τέλος Παράδειγμα

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ



Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει έναν αριθμό και να εκτυπώνει τον αντίστροφό του αριθμό.

ΑΛΓΟΡΙΘΜΟΣ

Αλγόριθμος Παράδειγμα

Διάβασε αριθμός

Αν αριθμός $\neq 0$ **τότε**

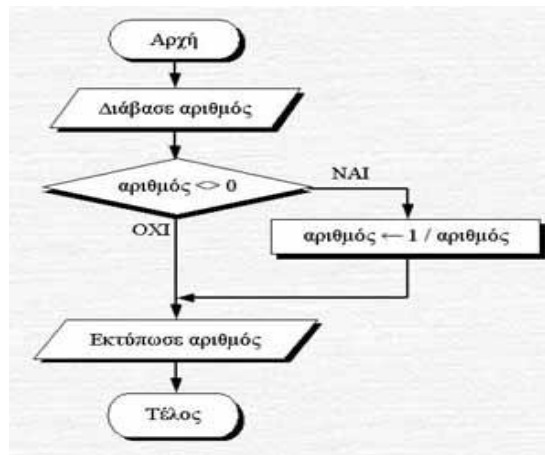
αριθμός $\leftarrow 1 / \text{αριθμός}$

Τέλος_Αν

Εκτύπωσε αριθμός

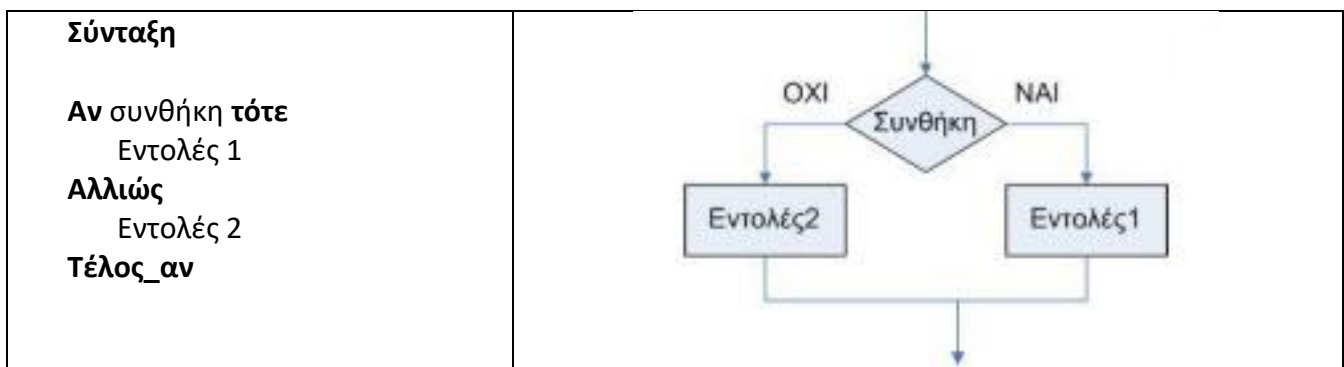
Τέλος Παράδειγμα

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ



2.4.2 Δομή Σύνθετης επιλογής

Η δομή σύνθετης επιλογής χρησιμοποιείται στην περίπτωση που επιθυμούμε να εκτελέσουμε εναλλακτικά δυο ομάδες εντολών. Κριτήριο για το ποιο σετ εντολών θα εκτελεστεί αποτελεί κάποια συνθήκη. Στην περίπτωση που η συνθήκη ισχύει θα εκτελεστεί η πρώτη ομάδα εντολών διαφορετικά θα εκτελεστεί η δεύτερη. Η γενική σύνταξη της σύνθετης δομής επιλογής είναι:



Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει δύο αριθμούς και να εκτυπωθεί ο μεγαλύτερος από τους δυο.

ΑΛΓΟΡΙΘΜΟΣ

Αλγόριθμος Παράδειγμα

Διάβασε αριθμός1, αριθμός2

Αν αριθμός1 < αριθμός2 **τότε**
αριθμός ← αριθμός2

Αλλιώς

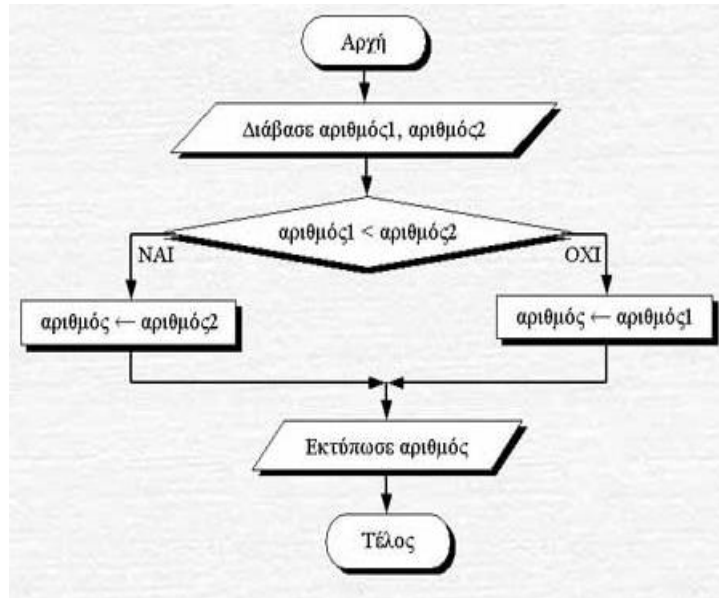
αριθμός ← αριθμός1

Τέλος_Αν

Εκτύπωσε αριθμός

Τέλος Παράδειγμα

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ



Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει έναν αριθμό και να εκτυπωθεί αντίστοιχο μήνυμα για το αν είναι άρτιος ή περιττός.

ΑΛΓΟΡΙΘΜΟΣ

Αλγόριθμος Παράδειγμα

Διάβασε αριθμός

Αν αριθμός mod 2 = 0 **τότε**
Εκτύπωσε "Άρτιος"

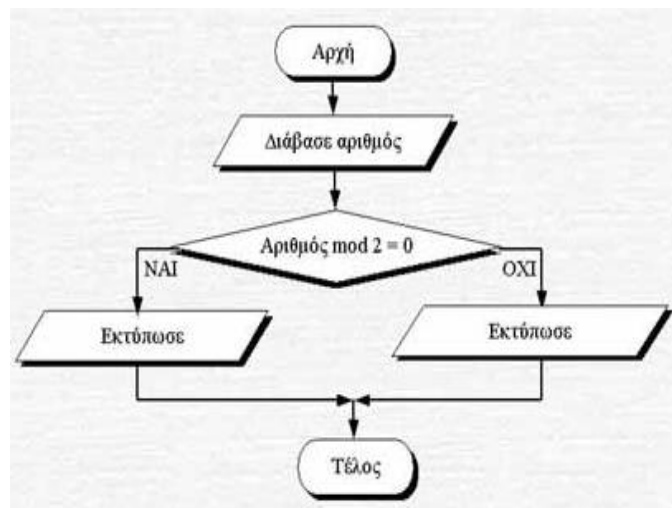
Αλλιώς

Εκτύπωσε "Περιττός"

Τέλος_Αν

Τέλος Παράδειγμα

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ



2.4.3 Δομές πολλαπλής επιλογής

Κάποια προβλήματα απαιτούν την επιλογή μεταξύ περισσότερων από δυο περιπτώσεις

A) **ΑΝ....ΑΛΛΙΩΣ_ΑΝ**

```

ΑΝ ΣΥΝΘΗΚΗ1 ΤΟΤΕ
    ΕΝΤΟΛΗ 1
    ...
    ΕΝΤΟΛΗ N
ΑΛΛΙΩΣ_ΑΝ ΣΥΝΘΗΚΗ2 ΤΟΤΕ
    ΕΝΤΟΛΗ 1
    ...
    ΕΝΤΟΛΗ N
ΑΛΛΙΩΣ
    ΕΝΤΟΛΗ 1
    ...
    ΕΝΤΟΛΗ N
ΤΕΛΟΣ_ΑΝ
    
```

B) **ΕΠΙΛΕΞΕ**

```

ΔΙΑΒΑΣΕ ΜΕΤΑΒΛΗΤΗ
ΕΠΙΛΕΞΕ ΜΕΤΑΒΛΗΤΗ
    ΠΕΡΙΠΤΩΣΗ ΣΥΝΘΗΚΗ1 Η ΤΙΜΗ Η ΑΠΟ..ΕΩΣ
        ΕΝΤΟΛΗ 1
        ....
        ΕΝΤΟΛΗ N
    ΠΕΡΙΠΤΩΣΗ ΣΥΝΘΗΚΗ2 Η ΤΙΜΗ Η ΑΠΟ..ΕΩΣ
        ΕΝΤΟΛΗ 1
        ....
        ΕΝΤΟΛΗ N
    ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
        ΕΝΤΟΛΗ 1
        ....
        ΕΝΤΟΛΗ N
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
    
```

Παράδειγμα : Να διαβαστεί ο μέσος όρος ενός μαθητή και να εκτυπωθεί χαρακτηρισμός του ΑΠΟΡΡΙΠΤΕΤΑΙ αν ο μέσος όρος του είναι μικρότερος του 9,5 , ΚΑΛΩΣ αν ο μέσος όρος είναι μικρότερος του 16, ΛΙΑΝ ΚΑΛΩΣ αν ο μέσος όρος του είναι μικρότερος του 18 , ΑΡΙΣΤΑ αν ο μέσος όρος του είναι μικρότερος από 20 και κάτω , ΛΑΘΟΣ ΔΕΔΟΜΕΝΑ αν ο μέσος όρος του είναι εκτός ορίων

Αλγόριθμος Παράδειγμα

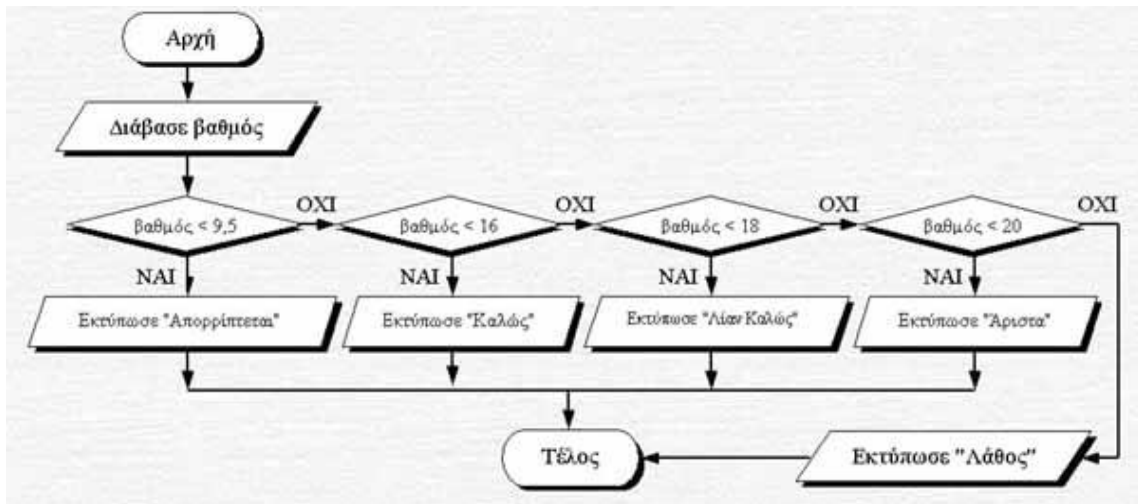
```

Διάβασε ΜΟ
Αν ΜΟ < 9,5 τότε
    Εκτύπωσε "Απορρίπτεται"
Αλλιώς_Αν ΜΟ < 16 τότε
    Εκτύπωσε "Καλώς"
Αλλιώς_Αν ΜΟ < 18 τότε
    Εκτύπωσε "Λίαν καλώς"
Αλλιώς_Αν ΜΟ < =20 τότε
    Εκτύπωσε "Άριστα"
Αλλιώς
    Εκτύπωσε "Λάθος δεδομένο"
Τέλος_Αν
Τέλος Παράδειγμα
    
```

Αλγόριθμος Παράδειγμα

```

Διάβασε ΜΟ
Επίλεξε ΜΟ
    Περίπτωση < 9,5
        Εκτύπωσε "Απορρίπτεται"
    Περίπτωση < 16
        Εκτύπωσε "Καλώς"
    Περίπτωση < 18
        Εκτύπωσε "Λίαν καλώς"
    Περίπτωση < =20
        Εκτύπωσε "Άριστα"
    Περίπτωση αλλιώς
        Εκτύπωσε "Λάθος δεδομένο"
Τέλος_επιλογών
Τέλος Παράδειγμα
    
```



Σχολιασμός:

- Μία από τις περιπτώσεις της δομής επιλογής εκτελούνται σε κάθε περίπτωση.
- Στη δομή **Επίλεξε** αν δεν ικανοποιείται καμία περίπτωση τότε εκτελείται το **Περίπτωση Αλλιώς**
- **ΠΡΟΣΟΧΗ:** Η **Περίπτωση Αλλιώς** δεν είναι υποχρεωτική.
- Τα σύμβολα ' και " χρησιμοποιούνται ισοδύναμα.
- Στις περιπτώσεις που θα ελεγχθούν περιλαμβάνουμε και αυτήν της λανθασμένη εισαγωγής δεδομένων.
- Τα παραπάνω θα μπορούσαν να υλοποιηθούν και με πολλές δομές σύνθετης επιλογής.

2.4.4 Εμφωλευμένες Δομές

Πολλαπλές επιλογές μπορούν να γίνουν και με μία εμφωλευμένη δομή. Το επόμενο παράδειγμα περιγράφει τον τρόπο με τον οποίο μία εντολή Αν... τότε είναι η εντολή που εκτελείται, όταν ισχύει (ή δεν ισχύει) η συνθήκη μίας άλλης εντολής Αν...τότε. Βέβαια η λογική αυτή μπορεί να επεκταθεί, δηλαδή να έχουμε νέα εμφωλευμένη δομή μέσα σε μία εμφωλευμένη δομή κ.ο.κ.

Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει τα έτη υπηρεσίας και τον μισθό ενός υπαλλήλου και να εκτυπωθεί το μόνους σύμφωνα με τον πίνακα:

Έτη υπηρεσίας	Μισθός	Μπόνους
< 12		–
	< 3500	15 %
	>= 3500	17 %

Αλγόριθμος Παράδειγμα

Διάβασε έτη_υπηρ, μισθ

Αν έτη_υπηρ < 12 **τότε**

μπόνους ← 0

Αλλιώς

Αν μισθός < 3500 **τότε**

μπόνους ← 0.15 * μισθός

Αλλιώς

μπόνους ← 0.17 * μισθός

Τέλος_Αν

Τέλος_Αν

Εκτύπωσε μπόνους

Τέλος Παράδειγμα

Παρατηρήσεις: Στο παραπάνω πρόβλημα οι δύο δομές επιλογής θα μπορούσαν να συγχωνευτούν σε μία, της οποίας η συνθήκη περιέχει και τις δύο παραπάνω συνθήκες. Για το σκοπό αυτό χρησιμοποιούνται οι συγκριτικοί τελεστές.

Πρόταση A	Πρόταση B	A Ή B	A ΚΑΙ B	ΟΧΙ A
Αληθής	Αληθής	Αληθής	Αληθής	Ψευδής
Αληθής	Ψευδής	Αληθής	Ψευδής	Ψευδής
Ψευδής	Αληθής	Αληθής	Ψευδής	Αληθής
Ψευδής	Ψευδής	Ψευδής	Ψευδής	Αληθής

Σχολιασμός πίνακα:

- Η πρόταση A Ή B είναι αληθής όταν κάποια από τις δύο ή και οι δύο είναι Αληθείς.
- Η πρόταση A ΚΑΙ B είναι αληθής όταν και οι δύο προτάσεις A, B είναι αληθείς.
- Η πρόταση ΟΧΙ A είναι αληθής όταν και η πρόταση A είναι ψευδής.

Η ιεραρχία των πράξεων των λογικών τελεστών είναι: ΟΧΙ, ΚΑΙ, Ή

Έτσι, ας υποθέσουμε ότι: **A=8 , B=2**

(A>=7)	ΚΑΙ (B=1)	Ψευδής
Αληθής	Ψευδής	
A < 20	Ή B < 4	Αληθής
αληθής	Αληθής	
A > B	ΚΑΙ B < 6	Αληθής
Αληθής	αληθής	
	ΟΧΙ B < 6	Ψευδής
	Αληθής	
A mod 2 = 0	Ή B mod 2 = 1	Αληθής
Αληθής	Ψευδής	

Σύμφωνα με τα παραπάνω ο αλγόριθμός του παραδείγματος τροποποιείται ως εξής:

Αλγόριθμος Παράδειγμα

Διάβασε έτη_υπηρ, μισθός

Αν έτη_υπηρεσίας < 12 **τότε**
μπόνους ← 0

Τέλος_Αν

Αν (έτη_υπηρ >= 12) **και** (μισθός < 3500) **τότε**
μπόνους ← 0.15 * μισθός

Τέλος_Αν

Αν (έτη_υπηρ >= 12) **και** (μισθός >= 3500) **τότε**

μπόνους $\leftarrow 0.17 * \text{μισθός}$

Τέλος_Αν

Εκτύπωσε μπόνους

Τέλος Παράδειγμα

2.4.5 Δομή Επανάληψης

Πολλές φορές πρέπει να εκτελεστεί μια ομάδα εντολών περισσότερες από μια φορές. Σ' αυτές τις περιπτώσεις χρησιμοποιούνται οι δομές επανάληψης. Υπάρχουν δυο περιπτώσεις: το πλήθος των επαναλήψεων να είναι γνωστό **ΕΞ ΑΡΧΗΣ** ή **ΌΧΙ**.

2.4.5.1 Η Δομή επανάληψης ΓΙΑ

ΣΥΝΤΑΞΗ

ΓΙΑ μεταβλητή **ΑΠΟ** Αρχική_τιμή **ΜΕΧΡΙ** Τελική_τιμή [**ΜΕ_ΒΗΜΑ** $\pm$ αριθμός]

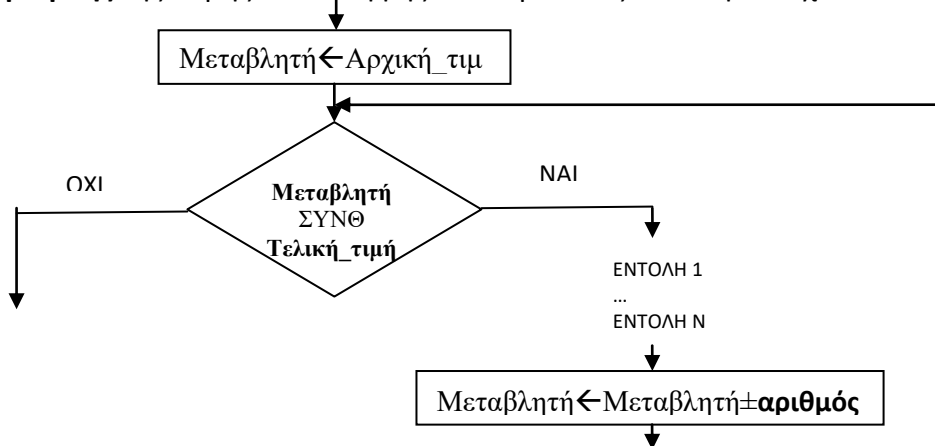
Εντολή1

...

Εντολή N

Τέλος_επανάληψης

Το διάγραμμα ροής της δομής επανάληψης **ΓΙΑ** παρουσιάζεται στη συνέχεια:



Όταν είναι ΓΝΩΣΤΟ ΤΟ ΠΛΗΘΟΣ των επαναλήψεων χρησιμοποιούμε την δομή επανάληψης **ΓΙΑ**

Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει **30 αριθμούς** και να εκτυπωθεί ο μέσος όρος τους.

Αλγόριθμος Παράδειγμα

άθροισμα $\leftarrow 0$

Για i από 1 μέχρι 30

Διάβασε αριθμός

 άθροισμα $\leftarrow$ άθροισμα + αριθμός

Τέλος_Επανάληψης

μέσος_όρος $\leftarrow$ άθροισμα / 30

Εκτύπωσε μέσος_όρος

Τέλος Παράδειγμα

Στην επαναληπτική δομή, οι εντολές που περιέχονται θα εκτελεστούν 30 φορές.

Μετά τον υπολογισμό του αθροίσματος η ροή του αλγορίθμου επανέρχεται στην εντολή **Για**.

Για i από 1 μέχρι 30

...

Τέλος_Επανάληψης

Η μεταβλητή i θα αυξηθεί από την τιμή 1 μέχρι την τιμή 30 (την ονομάζουμε μετρητή) κατά μία μονάδα κάθε φορά..

Αφού διαβαστεί κάθε ένας αριθμός από τους 30 και εισχωρηθεί στην ομώνυμη μεταβλητή, προστίθεται στη μεταβλητή.

Το τμήμα εντολών που βρίσκεται εντός μιας επανάληψης ονομάζεται **ΒΡΟΧΟΣ**.

Πρέπει επίσης, να τονιστεί ότι η εντολή **άθροισμα $\leftarrow$ άθροισμα + αριθμός**, προσθέτει το περιεχόμενο της μεταβλητής **άθροισμα** με αυτό της **αριθμός** και τοποθετεί το αποτέλεσμα στην μεταβλητή άθροισμα.

Ένα σύνηθες λάθος των μαθητών είναι να γράφουν λανθασμένα **άθροισμα $\leftarrow$ αριθμός**.

Με τη χρήση της εντολής αυτής όμως, θα καταχωρηθεί στη μεταβλητή άθροισμα ο τελευταίος αριθμός που διαβάστηκε. Πρέπει λοιπόν, στην τρέχουσα τιμή της μεταβλητής άθροισμα να προστίθεται η νέα τιμή.

Επιπρόσθετα, πρέπει πάνω από την επαναληπτική δομή να μηδενίσουμε την μεταβλητή άθροισμα, διαφορετικά δεν μπορούμε να γνωρίζουμε την τιμή της κατά την πρώτη επανάληψη.

Γενικώς, **δεν μπορούμε να χρησιμοποιήσουμε μια μεταβλητή, αν προηγούμενα δεν έχει λάβει τιμή – αρχικοποίηση**. Η αρχικοποίηση μπορεί να πραγματοποιηθεί είτε με την είσοδο δεδομένων σε μια μεταβλητή (εντολή **Διάβασε**) είτε με **εκχώρηση τιμής**.

Παράδειγμα : Να βρεθούν και να εκτυπωθούν οι διψήφιοι άρτιοι αριθμοί.

Αλγόριθμος Παράδειγμα

Για i από 10 μέχρι 99 με_βήμα 2

Εκτύπωσε i

Τέλος_Επανάληψης

Τέλος Παράδειγμα

Η συμπλήρωση της εντολής **Για με το βήμα να είναι 2** αυξάνει την τιμή της μεταβλητής i αντίστοιχα ώστε με αρχική τιμή την τιμή 10 να λαμβάνει τις τιμές 10, 12, 14, 16, 18, 20, ..., 94, 96, 98.

Ο επόμενος άρτιος, το 100, δεν είναι διψήφιος και το i δεν θα λάβει την τιμή αυτή.

Σημείωση 1: αν επιθυμούσαμε φθίνουσα διάταξη στην εκτύπωση των αριθμών ο βρόχος θα είχε τη μορφή

Για i από 99 μέχρι 10 με_βήμα -2

Εκτύπωσε i

Τέλος_Επανάληψης

Όπου η μεταβλητή i θα λάβει τις τιμές **99, 97, 95, ..., 17, 15, 13, 11**.

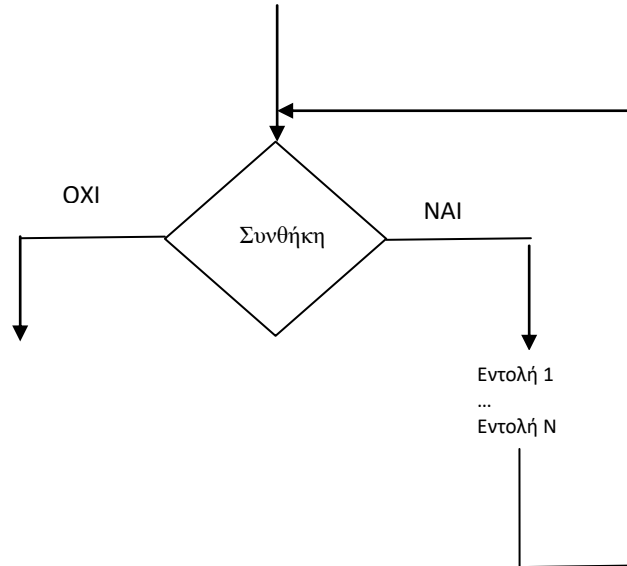
Σημείωση: Η τιμή του βήματος δεν είναι απαραίτητο να είναι ακέραια τιμή, αλλά μπορεί να λάβει και δεκαδική τιμή.

2.4.5.2 Η Δομή επανάληψης Όσο...Επανάλαβε

ΣΥΝΤΑΞΗ

Το **διάγραμμα ροής** της δομής επανάληψης **Όσο...Επανάλαβε** παρουσιάζεται στη συνέχεια:

Όσο Συνθήκη **Επανάλαβε**
 Εντολή 1
 ...
 Εντολή N
Τέλος_επανάληψης



Όταν **ΔΕΝ ΕΙΝΑΙ ΚΑΘΟΡΙΣΜΕΝΟ** το πλήθος των επαναλήψεων εξ'αρχής τότε δεν μπορεί να χρησιμοποιηθεί η δομή επανάληψης **ΓΙΑ** που παρουσιάστηκε στην προηγούμενη παράγραφο.

Εφόσον μετά από κάθε επανάληψη ελέγχεται εκ νέου η συνθήκη, πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μία εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Σε αντίθετη περίπτωση η επανάληψη δε θα τερματίζεται και θα εκτελείται συνεχώς.

Η χρήση τιμών για τον τερματισμό μίας επαναληπτικής διαδικασίας, όπως στο παράδειγμα η αυθαίρετη επιλογή του 0, είναι συνήθης στον προγραμματισμό.

Η τιμή αυτή ορίζεται από τον προγραμματιστή και αποτελεί μια σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή είναι τέτοια, ώστε να μην είναι λογικά σωστή για το πρόβλημα, για παράδειγμα η τιμή 0 αποκλείεται από τις μετρήσεις σύμφωνα με την εκφώνηση του παραδείγματος.

Η τιμή αυτή συχνά αποκαλείται "τιμή φρουρός".

Σε αυτήν την περίπτωση χρησιμοποιείται η δομή **Όσο...επανάλαβε**, η οποία θα περιγραφεί με τη βοήθεια ενός παραδείγματος.

Παράδειγμα: Ο ταμίας σε ένα Super Market καταχωρεί στην ταμειακή απόδειξη τον κωδικό του προϊόντος και την τιμή. Όταν ολοκληρωθεί η καταχώρηση για την αγορά ενός πελάτη, τότε εισάγεται ο κωδικός 0.

Να εκτυπωθεί το συνολικό ποσό της αγοράς και το πλήθος των προϊόντων που αγοράστηκαν.

Για την επίλυση του προβλήματος πρέπει να διαβάζονται οι κωδικοί και οι τιμές των προϊόντων μέχρι την εισαγωγή του **κωδικού 0**.

Αλγόριθμος Παράδειγμα

σύνολο $\leftarrow 0$

πλήθος $\leftarrow 0$

Διάβασε κωδικός

Όσο κωδικός $\neq 0$ **επανάλαβε**

Διάβασε τιμή, τεμάχια

σύνολο $\leftarrow$ σύνολο + τιμή * τεμάχια

πλήθος $\leftarrow$ πλήθος + 1

Διάβασε κωδικός

Τέλος_Επανάληψης

Εκτύπωσε σύνολο, πλήθος

Τέλος Παράδειγμα

Ο βρόχος επαναλαμβάνεται για άγνωστο αριθμό επαναλήψεων.

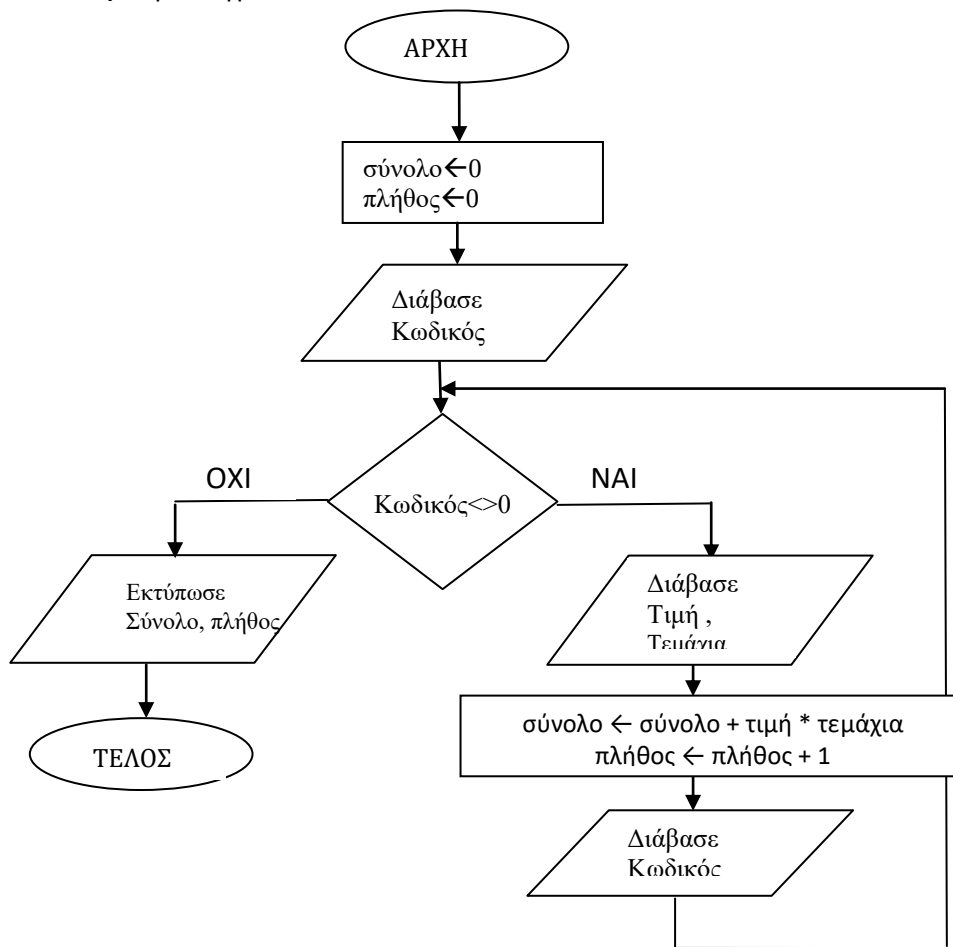
Η ροή εκτέλεσης μετά την εντολή

Τέλος_επανάληψης επανέρχεται στην εντολή **Όσο...Επανάλαβε**

Όσο κωδικός $\neq 0$ **επανάλαβε**

...

Τέλος_Επανάληψης



Η μεταβλητή σύνολο περιέχει το σύνολο της αγοράς μέχρι την τρέχουσα επανάληψη. Είναι σαφές ότι δεν είναι γνωστό το πλήθος των προϊόντων που αγοράστηκαν.

Για το λόγο αυτό χρησιμοποιείται η μεταβλητή **πλήθος**. Ένα σύνηθες λάθος που κάνουν οι μαθητές είναι ότι ξεχνούν την εντολή **Διάβασε** κωδικός πριν το βρόχο. Ωστόσο, αν δεν διαβαστεί η μεταβλητή κωδικός ώστε να λάβει τιμή, πώς θα την χρησιμοποιήσουμε στη συνθήκη του **Όσο**; Στο εσωτερικό του βρόχου διαβάζουμε και τις μεταβλητές τιμή και τεμάχια που αφορούν το προϊόν και προσθέτουμε το γινόμενο τους στη μεταβλητή σύνολο (το οποίο έχει αρχικοποιηθεί). Αυξάνουμε δε, τη μεταβλητή πλήθος κατά μία μονάδα. Τέλος, διαβάζεται ο κωδικός του νέου προϊόντος, ώστε να ελεγχθεί κατά την νέα επανάληψη.

Πρέπει να τονιστεί ότι ο έλεγχος της συνθήκης στη δομή Όσο ... επανάλαβε πραγματοποιείται στην αρχή του βρόχου.

2.4.5.3 Η Δομή επανάληψης Αρχή_Επανάληψης...Μέχρις_Ότου

Συμπληρωματικά της δομής Όσο...επανάλαβε χρησιμοποιείται η δομή **Αρχή_επανάληψης...Μέχρις_Ότου**.

Η σύνταξη της δομής είναι:

ΣΥΝΤΑΞΗ

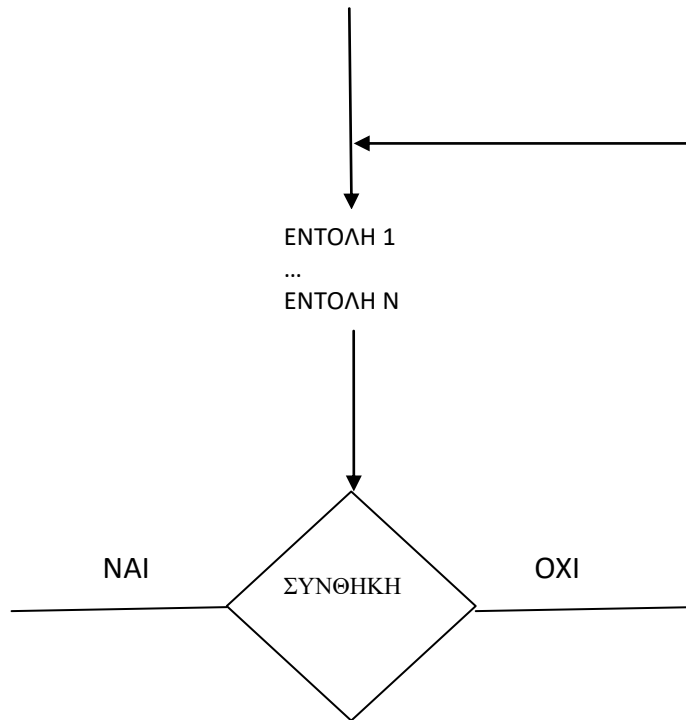
Αρχή_Επανάληψης

Εντολή1

.....

ΕντολήN

Μέχρις_Ότου <συνθήκη>



Παρατηρούμε ότι σε αυτή τη δομή ο έλεγχος της συνθήκης πραγματοποιείται στο **τέλος του βρόχου**.

Εκτελούνται οι εντολές μεταξύ των **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**. Στη συνέχεια ελέγχεται η λογική έκφραση και αν δεν ισχύει (είναι ψευδής), τότε οι εντολές που βρίσκονται ανάμεσα στις **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**, εκτελούνται πάλι.

Ελέγχεται ξανά η λογική έκφραση και αν δεν ισχύει, επαναλαμβάνεται η εκτέλεση των ίδιων εντολών. Όταν η λογική έκφραση γίνει Αληθής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά από την **ΜΕΧΡΙΣ_ΟΤΟΥ**.

Πολύ συχνά η ίδια επαναληπτική διαδικασία μπορεί να γραφεί εξίσου σωστά χρησιμοποιώντας είτε τη δομή **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ** είτε τη δομή **ΜΕΧΡΙΣ_ΟΤΟΥ** και είναι προσωπική επιλογή του προγραμματιστή ποια από τις δυο θα χρησιμοποιήσει.

Υπάρχουν όμως περιπτώσεις όπου η χρήση της εντολής **ΜΕΧΡΙΣ_ΟΤΟΥ** οδηγεί σε απλούστερα και πιο ευκολονόητα προγράμματα. Γενικά σε περιπτώσεις όπου η επανάληψη θα συμβεί υποχρεωτικά μία φορά, είναι προτιμότερη η χρήση της **ΜΕΧΡΙΣ_ΟΤΟΥ**. Χαρακτηριστική περίπτωση όπου προτιμάται η εντολή **ΜΕΧΡΙΣ_ΟΤΟΥ** είναι στον έλεγχο αποδεκτών τιμών καθώς και στην επιλογή από προκαθορισμένες απαντήσεις ή μενού.

Παράδειγμα: Να γραφεί αλγόριθμος που να διαβάζει αριθμούς από το πληκτρολόγιο, να υπολογίζει το πλήθος των περιττών αριθμών και να εκτυπώνει το μέσο όρο τους. Η διαδικασία ανάγνωσης να σταματά όταν διαβαστούν 20 περιττοί αριθμοί.

Αλγόριθμος Παράδειγμα

άθροισμα $\leftarrow$ 0

πλήθος $\leftarrow$ 0

Αρχή_Επανάληψης

Διάβασε αριθμός

Αν αριθμός mod 2 = 1 τότε

 άθροισμα $\leftarrow$ άθροισμα + αριθμός

 πλήθος $\leftarrow$ πλήθος + 1

Τέλος_Αν

Μέχρις_Ότου πλήθος = 20

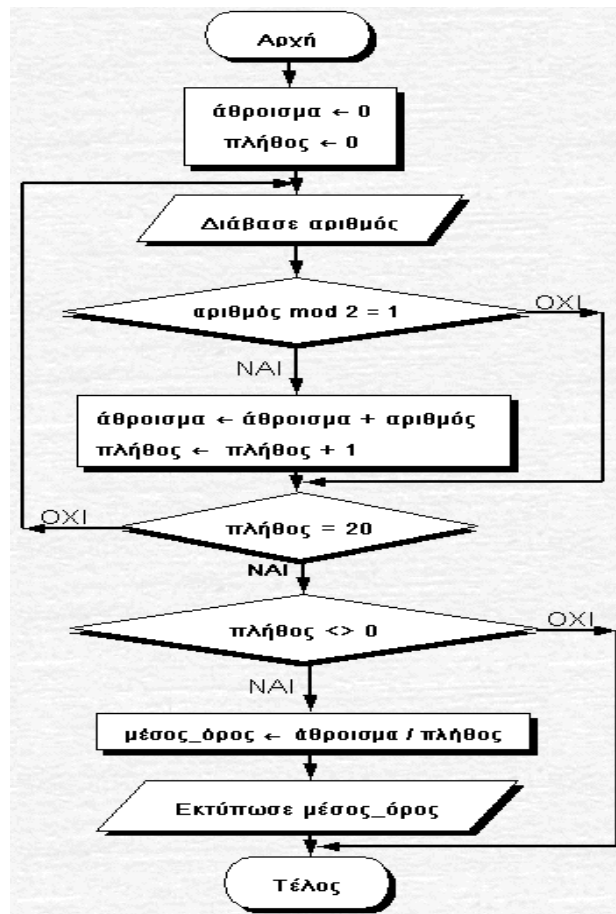
Αν πλήθος $\neq$ 0 τότε

 μέσος_όρος $\leftarrow$ άθροισμα / πλήθος

Εκτύπωσε μέσος_όρος

Τέλος_Αν

Τέλος Παράδειγμα



ΠΑΡΑΤΗΡΗΣΕΙΣ ΓΙΑ ΤΙΣ ΔΟΜΕΣ ΕΠΑΝΑΛΗΨΗΣ

Η δομή επανάληψης **Για** χρησιμοποιείται όταν είναι γνωστό το πλήθος των επαναλήψεων ενώ οι δομές **Όσο...επανάλαβε** και **Αρχή_Επανάληψης...Μέχρις_Ότου** στις άλλες περιπτώσεις.

Ένα ζήτημα που μπορεί να μας απασχολήσει στις ασκήσεις είναι η μετατροπή από την μία δομή στην άλλη. Μια δομή **ΓΙΑ** πάντα μπορεί να μετατραπεί, στις **Όσο...επανάλαβε** και **Αρχή_Επανάληψης ... Μέχρις_Ότου**, το αντίθετο σε **ΓΙΑ** δεν ισχύει (**πάντα**).

Η δομή **Όσο...επανάλαβε** μετατρέπεται πάντα στην **Αρχή_Επανάληψης ... Μέχρις_Ότου**.

Ας υποθέσουμε ότι έχουμε το εξής τμήμα αλγορίθμου σε δομή επανάληψης **Για**:

$\text{άθροισμα} \leftarrow 0$

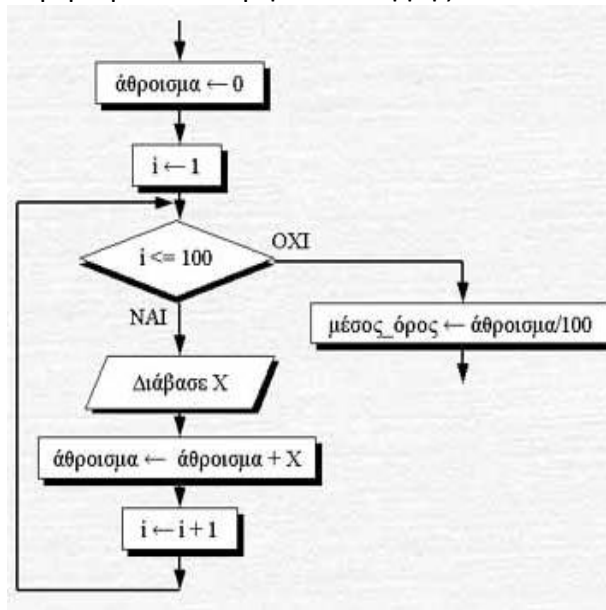
Για i από 1 μέχρι 100

Διάβασε X

$\text{άθροισμα} \leftarrow \text{άθροισμα} + X$

Τέλος_Επανάληψης

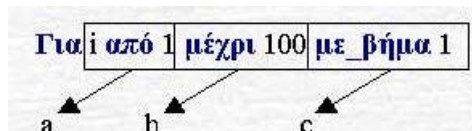
$\text{μέσος_όρος} \leftarrow \text{άθροισμα} / 100$



Παρατηρούμε ότι στο διάγραμμα ροής δεν υπάρχει ειδικό σύμβολο για τη δομή **Για**. Η υλοποίησή της απαιτεί τη χρήση μιας σειράς εντολών. Αυτές είναι:

- Η εντολή εκχώρησης $i \leftarrow 1$, για αρχικοποίηση της μεταβλητής
- Η συνθήκη τερματισμού $i \leq 100$
- Η εντολή εκχώρησης $i \leftarrow i + 1$

Και οι τρεις παραπάνω εντολές περιέχονται στην δομή **Για**



Από το διάγραμμα ροής του προηγούμενου παραδείγματος είναι πολύ εύκολο να δημιουργήσουμε τον ίδιο βρόχο με τις δομές **Όσο...επανάλαβε** και **Αρχή_Επανάληψης ... Μέχρις_Ότου**.

$\text{άθροισμα} \leftarrow 0$
 $i \leftarrow 1$

Όσο $i \leq 100$ **επανάλαβε**

Διάβασε X

$\text{άθροισμα} \leftarrow \text{άθροισμα} + X$

$i \leftarrow i + 1$

Τέλος_Επανάληψης

$\text{μέσος_όρος} \leftarrow \text{άθροισμα} / 100$

$\text{άθροισμα} \leftarrow 0$
 $i \leftarrow 1$

Αρχή_Επανάληψης

Διάβασε X

$\text{άθροισμα} \leftarrow \text{άθροισμα} + X$

$i \leftarrow i + 1$

Μέχρις_Ότου $i > 100$

$\text{μέσος_όρος} \leftarrow \text{άθροισμα} / 100$

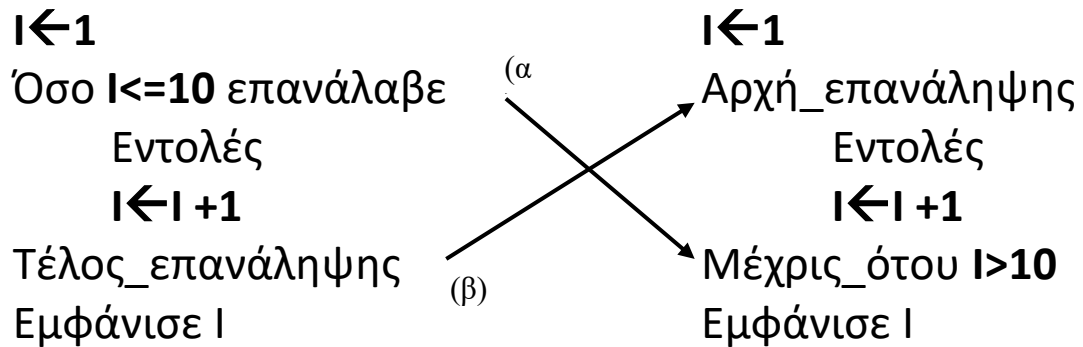
Παρατηρούμε ότι οι συνθήκες των δύο δομών είναι **ΑΝΤΙΘΕΤΕΣ** ενώ το τμήμα εντολών του βρόχου παραμένουν το ίδιο.

ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ			
Στις 3εις Δομές (ΓΙΑ ⇔ ΟΣΟ ⇔ ΜΕΧΡΙΣ_ΟΤΟΥ)			
	ΓΙΑ	ΟΣΟ	ΜΕΧΡΙΣ_ΟΤΟΥ
	ΓΙΑ ΜΕΤ ΑΠΟ Α_Τ ΜΕΧΡΙ Τ_Τ [ΜΕ_ΒΗΜΑ ± ΑΡ] ΕΝΤΟΛΕΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΜΕΤ←Α_Τ ΟΣΟ ΜΕΤ ΣΥΝΘΗΚΗ Τ_Τ ΕΠΑΝΑΛΑΒΕ ΕΝΤΟΛΕΣ ΜΕΤ←ΜΕΤ ± ΑΡ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΜΕΤ←Α_Τ ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΕΝΤΟΛΕΣ ΜΕΤ←ΜΕΤ ± ΑΡ ΜΕΧΡΙΣ_ΟΤΟΥ ΜΕΤ ΣΥΝΘΗΚΗ Τ_Τ
1	Για Ι από 1 μέχρι 10 Εντολές Τέλος_επανάληψης	Ι←1 Όσο Ι≠10 επανάλαβε Εντολές Ι←Ι+1 Τέλος_επανάληψης	Ι←1 Αρχή_επανάληψης Εντολές Ι←Ι+1 Μέχρις_ότου Ι>10
2	Για Ι από 1 μέχρι 10 με βήμα 2 Εντολές Τέλος_επανάληψης	Ι←1 Όσο Ι≠10 επανάλαβε Εντολές Ι←Ι+2 Τέλος_επανάληψης	Ι←1 Αρχή_επανάληψης Εντολές Ι←Ι+2 Μέχρις_ότου Ι>10
3	Για Ι από 10 μέχρι 1 με_βήμα -0.5 Εντολές Τέλος_επανάληψης	Ι←10 Όσο Ι>=1 επανάλαβε Εντολές Ι←Ι - 0.5 Τέλος_επανάληψης	Ι←10 Αρχή_επανάληψης Εντολές Ι←Ι - 0.5 Μέχρις_ότου Ι<1

Στις 2 Δομές (ΟΣΟ $\leftrightarrow$ ΜΕΧΡΙΣ_ΟΤΟΥ)

4	ΔΙΑΒΑΣΕ ΜΕΤΑΒΛ ΌΣΟ ΜΕΤΑΒΛ <> 'ΑΛΦΑΡΙΘΜΗΤΙΚΟ' ΕΠΑΝΑΛΑΒΕ ΕΝΤΟΛΕΣ ΔΙΑΒΑΣΕ ΜΕΤΑΒΛ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΔΙΑΒΑΣΕ ΜΕΤΑΒΛ ΑΝ ΜΕΤΑΒΛ <> 'ΑΛΦΑΡΙΘΜΗΤΙΚΟ' ΤΟΤΕ ΕΝΤΟΛΕΣ ΤΕΛΟΣ_ΑΝ ΜΕΧΡΙΣ_ΟΤΟΥ ΜΕΤΑΒΛ='ΑΛΦΑΡΙΘΜΗΤΙΚΟ'
5	ΔΙΑΒΑΣΕ ΜΕΤΑΒΛ ΟΣΟ ΜΕΤΑΒΛ <> ΑΡΙΘΜΟΣ ΕΠΑΝΑΛΑΒΕ ΕΝΤΟΛΕΣ ΔΙΑΒΑΣΕ ΜΕΤΑΒΛ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΔΙΑΒΑΣΕ ΜΕΤΑΒΛ ΑΝ ΜΕΤΑΒΛ <> ΑΡΙΘΜΟΣ ΤΟΤΕ ΕΝΤΟΛΕΣ ΤΕΛΟΣ_ΑΝ ΜΕΧΡΙΣ_ΟΤΟΥ ΜΕΤΑΒΛ = ΑΡΙΘΜΟΣ
6	ΜΕΤΑΒΛ $\leftarrow$ 0 ΟΣΟ ΜΕΤΑΒΛ ΣΥΝΘ ΑΡΙΘΜΟΣ ΕΠΑΝΑΛΑΒΕ ΕΝΤΟΛΕΣ ΜΕΤΑΒΛ $\leftarrow$ ΜΕΤΑΒΛ + ΑΡΙΘΜ_ΤΙΜΗ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	ΜΕΤΑΒΛ $\leftarrow$ 0 ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΕΝΤΟΛΕΣ ΜΕΤΑΒΛ $\leftarrow$ ΜΕΤΑΒΛ + ΑΡΙΘΜ_ΤΙΜΗ ΜΕΧΡΙΣ_ΟΤΟΥ ΜΕΤΑΒΛ ΟΧΙ (ΣΥΝΘ) ΑΡΙΘΜΟΣ

ΕΣΤΩ Η ΕΠΑΝΑΛΗΨΗ ΟΣΟ

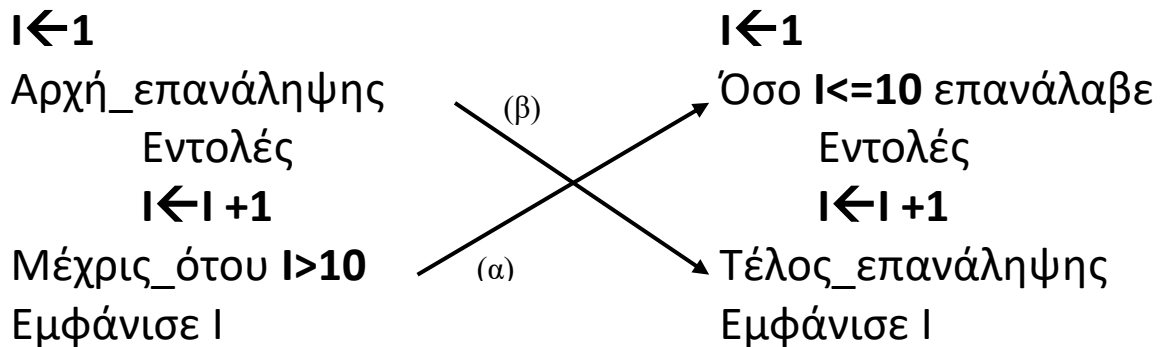


Σύμφωνα με το παραπάνω έχουμε τα εξής βήματα :

1. Οι εντολές πριν και μετά την **ΟΣΟ** παραμένουν ως έχουν στην **ΜΕΧΡΙΣ_ΟΤΟΥ**.
2. Η **ΟΣΟ** συνθήκη σε **ΜΕΧΡΙΣ_ΟΤΟΥ** συνθήκη (α) και ακολουθούμε την διαδικασία κανόνων (μετατροπής βλέπε παρακάτω)
3. Το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ** της **ΟΣΟ** σε **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** της **ΜΕΧΡΙΣ_ΟΤΟΥ** (β)

ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ ΑΠΟ ΜΕΧΡΙΣ_ΟΤΟΥ σε ΟΣΟ

ΕΣΤΩ Η ΕΠΑΝΑΛΗΨΗ ΜΕΧΡΙΣ_ΟΤΟΥ



Σύμφωνα με το παραπάνω έχουμε τα εξής βήματα :

1. Οι εντολές πριν και μετά την **ΜΕΧΡΙΣ_ΟΤΟΥ** παραμένουν ως έχουν στην **ΟΣΟ**
2. Η **ΜΕΧΡΙΣ_ΟΤΟΥ** συνθήκη σε **ΟΣΟ** συνθήκη (α) και ακολουθούμε την διαδικασία κανόνων μετατροπής βλέπε παρακάτω.
3. Η **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** της **ΜΕΧΡΙΣ_ΟΤΟΥ** σε **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ** της **ΟΣΟ** (β)

Οι βασικοί συγκριτικοί τελεστές είναι: $<$, $>$, $=$

ΟΣΟ συνθήκη	ΜΕΧΡΙΣ_ΟΤΟΥ συνθήκη
$<$	$>=$
$>$	$<=$
$=$	$<>$
$<=$	$>$
$>=$	$<$
$<>$	$=$

Έτσι σύμφωνα με τον παραπάνω πίνακα έχουμε τα εξής:

Αν σε μια δομή επανάληψης πχ **ΟΣΟ** έχουμε συνθήκη με συγκριτικό τελεστή έναν από τους τρεις βασικούς τελεστές (πχ $<$) τότε η άλλη δομή επανάληψης **ΜΕΧΡΙΣ_ΟΤΟΥ** θα έχει τους άλλους δύο (δηλ . $>=$).

Παραδείγματα

1. Δίνεται η παρακάτω ακολουθία εντολών. Να γραφούν Ισοδύναμα χρησιμοποιώντας την **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ**.

A)

```
I ← 1
ΟΣΟ I <= 10 ΕΠΑΝΑΛΑΒΕ
  ΔΙΑΒΑΣΕ B
  I ← I+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
I ← 100
ΟΣΟ I > 1 ΕΠΑΝΑΛΑΒΕ
  ΔΙΑΒΑΣΕ B
  I ← I-1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

2. Δίνεται η παρακάτω ακολουθία εντολών. Να γίνει ισοδύναμο τμήμα αλγορίθμου με την **ΟΣΟ**.

```
I ← 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ B
  I ← I+1
ΜΕΧΡΙΣ_ΟΤΟΥ I > 10
```

3. Δίνεται η παρακάτω ακολουθία εντολών. Να γίνει ισοδύναμος αλγόριθμος με την **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΟΣΟ**.

A)

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 3
  ΔΙΑΒΑΣΕ B
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
I ← 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ B
  I ← I+1
ΜΕΧΡΙΣ_ΟΤΟΥ I > 10
```

Γ)

```
ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 1
  ΔΙΑΒΑΣΕ B
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

4. Δίνεται η παρακάτω ακολουθία εντολών. Να γίνει ισοδύναμο τμήμα αλγορίθμου με την ΓΙΑ.

A)

```
I ← 1
ΟΣΟ I ≤ 10 ΕΠΑΝΑΛΑΒΕ
  ΔΙΑΒΑΣΕ B
  I ← I+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

B)

```
I ← -10
ΟΣΟ I ≤ 10 ΕΠΑΝΑΛΑΒΕ
  ΔΙΑΒΑΣΕ B
  I ← I+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

5. Δίνεται η παρακάτω ακολουθία εντολών. Να γίνει ισοδύναμο τμήμα αλγορίθμου με την ΓΙΑ.

```
ΑΝ I ≤ 10 ΤΟΤΕ
  ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ B
    I ← I+1
  ΜΕΧΡΙΣ_ΟΤΟΥ I > 10
ΤΕΛΟΣ_ΑΝ
```

1. Καταγράφουμε μία προς μία τις μεταβλητές του αλγορίθμου ή προγράμματος ή τμημάτων αυτών που μας δίνουν από μία φορά άσχετα με το πόσες φορές επαναλαμβάνονται και τις βάζουμε σε στήλες την μία μεταβλητή δίπλα στην άλλη.
2. Όταν υπάρχει εντολή **εξόδου** τότε στο τέλος των μεταβλητών γράφουμε μία επιπλέον στήλη με όνομα (**Αποτελέσματα ή Έξοδος ή Οθόνη**).
3. Σε κάθε μεταβλητή βάζουμε την τιμή της κατά την εκτέλεση του αλγορίθμου ή προγράμματος ή τμημάτων αυτών.
4. Αλλάζουμε γραμμή όταν αλλάζει η τιμή μιας μεταβλητής.

Παραδείγματα

A)

```
X ← 10
Y ← 95
AN X < 19 ΤΟΤΕ
    X ← X + 2
    Y ← Y - X + 1
    ΕΜΦΑΝΙΣΕ X + Y
ΤΕΛΟΣ_ΑΝ
```

X	Y	ΕΞΟΔΟΣ
10	95	
12	84	96

B)

```
X ← 10
Y ← 95
AN X < 19 ΤΟΤΕ
    X ← X + 2
    Y ← Y - X + 1
    ΕΜΦΑΝΙΣΕ X + Y
ΤΕΛΟΣ_ΑΝ
```

X	Y	ΕΞΟΔΟΣ
10		
	95	
12		
	84	
		96

- 1. Καταγράφουμε μία προς μία** τις μεταβλητές του αλγορίθμου ή προγράμματος ή τμημάτων αυτών που μας δίνουν από μία φορά άσχετα με το πόσες φορές επαναλαμβάνονται και τις βάζουμε σε στήλες την μία μεταβλητή δίπλα στην άλλη.
- 2. Όταν υπάρχει εντολή εξόδου** τότε στο τέλος των μεταβλητών βάζουμε μία επιπλέον στήλη με όνομα **(Αποτελέσματα ή Έξοδος ή Οθόνη)**.
- 3. Αριστερά από τις μεταβλητές** βάζουμε μία επιπλέον στήλη με όνομα **(Αριθμός Εντολής)**
- 4. Σε κάθε μεταβλητή** βάζουμε την τιμή της κατά την εκτέλεση του αλγορίθμου ή προγράμματος ή τμημάτων αυτών καθώς και στην στήλη Αριθμός Εντολής τον αριθμό της γραμμής της εντολής.

Ακολουθεί παράδειγμα

A)

```

1 X ← 10
2 Y ← 95
3 AN X < 19 TOTE
4   X ← X + 2
5   Y ← Y - X + 1
6   ΕΜΦΑΝΙΣΕ X + Y
7 ΤΕΛΟΣ_ΑΝ

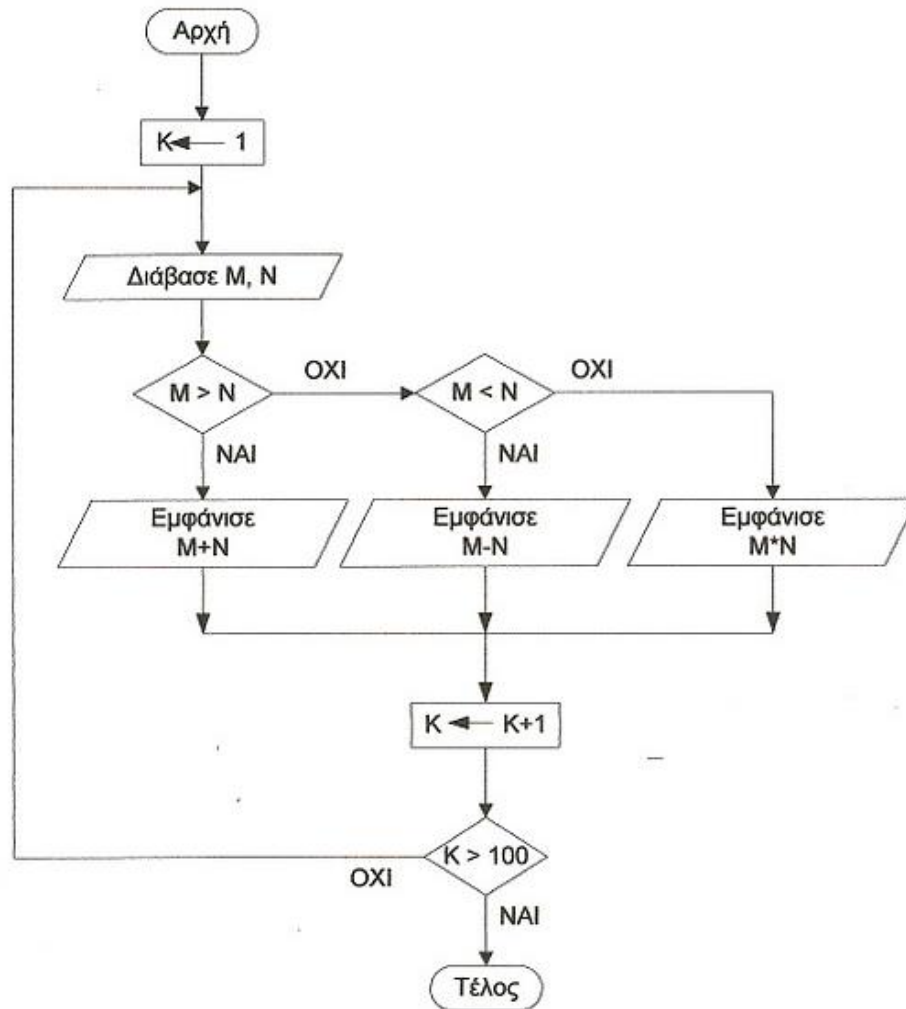
```

ΑΡ.ΕΝΤΟΛΗΣ	X	Y	ΕΞΟΔΟΣ
1	10		
2		95	
4	12		
5		84	
6			96

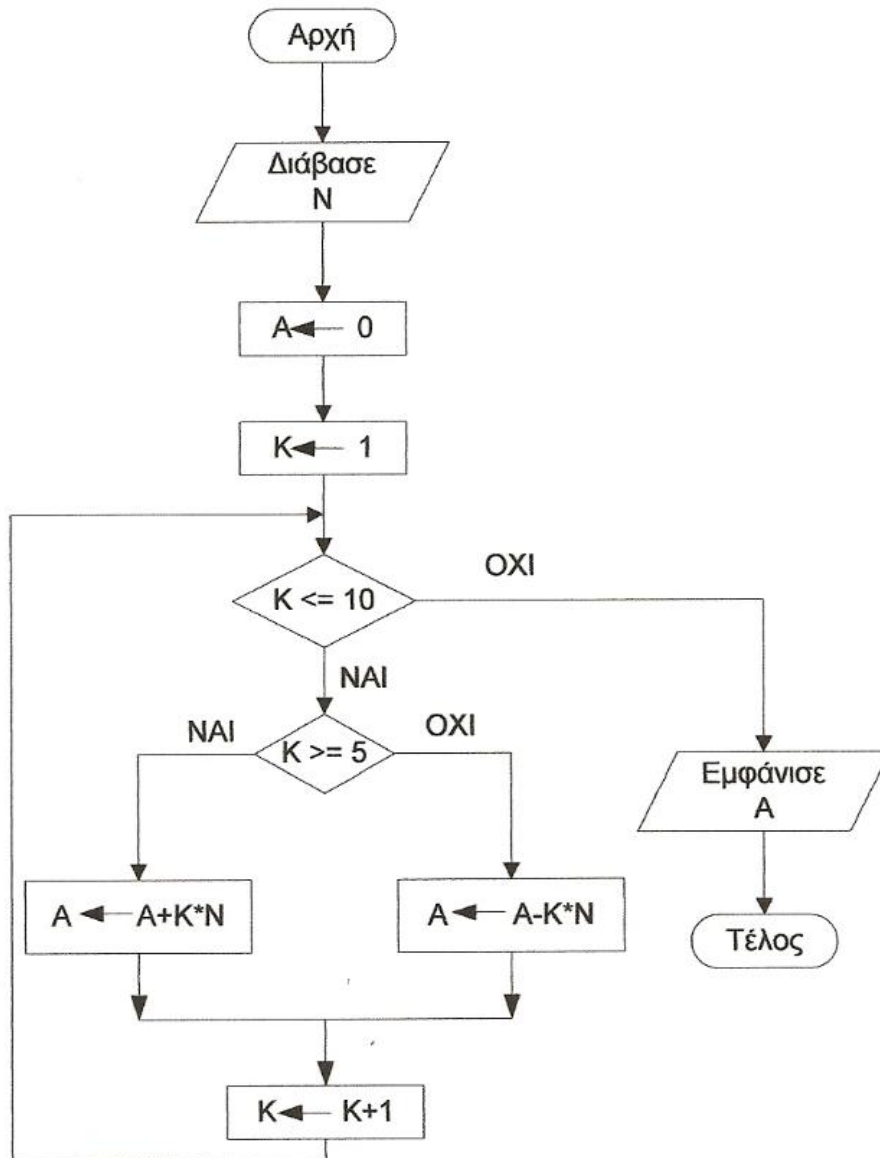
1. Βλέπω στο διάγραμμα πόσους ρόμβους έχω.
2. Αποφασίζω αν οι ρόμβοι είναι επιλογή ή επανάληψη.
3. Στην επιλογή δεν έχουμε κάποιο βελάκι που γυρίζει προ τα πίσω (δηλαδή προς τον Ρόμβο), ενώ στην επανάληψη έχουμε.
4. Στην επιλογή, βλέπουμε αν από το ρόμβο που φεύγουν 2 γραμμές (Αληθής ή Ψευδής) υπάρχουν εντολές στο Ψευδές για να γράψουμε τις εντολές στο **Αλλιώς**.
5. Εφόσον αποφασίσουμε για το ρόμβο ότι είναι επανάληψη, πρέπει να αποφασίσουμε τώρα αν είναι **ΟΣΟ** ή **ΜΕΧΡΙΣ_ΟΤΟΥ**.
6. Όταν είναι **ΟΣΟ**, το βελάκι που γυρίζει πίσω, είναι πάντα ακριβώς πριν το ρόμβο της επανάληψης. **(σχεδόν πάντα)**
7. Όταν είναι **ΜΕΧΡΙΣ_ΟΤΟΥ**, το βελάκι είναι πριν από κάποιες (δεν μας ενδιαφέρει πόσες) εντολές και ύστερα ακολουθεί ο ρόμβος. Εκεί που υπάρχει το βελάκι προς τα πίσω , στο ψευδοκώδικα γράφουμε **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ**.
8. Τέλος πρέπει να ελέγξουμε πώς φεύγει η κάθε επανάληψη. Στην **ΟΣΟ** πρέπει να φεύγει με **ΨΕΥΔΗΣ** ενώ να γυρίζει με **ΑΛΗΘΗΣ**. Αν συμβαίνει το αντίθετο (**να φεύγει με ΑΛΗΘΗΣ**) τότε αυτό που πρέπει να κάνουμε είναι να αλλάξουμε την συνθήκη μέσα στο ρόμβο και να την αντικαταστήσουμε με **ΟΧΙ**. **(παράδειγμα : $\alpha > 0$ γίνεται $\alpha \leq 0$).**
9. Αν στη **ΜΕΧΡΙΣ_ΟΤΟΥ** φεύγει με **ΨΕΥΔΗΣ** και γυρίζει με **ΑΛΗΘΗΣ** τότε πρέπει στη συνθήκη μέσα στο ρόμβο να την αλλάξουμε και να τοποθετήσουμε την **ΟΧΙ**.

Να μετατρέψετε τα παρακάτω Λογικά Διαγράμματα σε ισοδύναμους ΨΕΥΔΟΚΩΔΙΚΕΣ.

A)



B)



2.5 ΟΛΙΣΘΗΣΗ ΔΕΞΙΑ – ΑΡΙΣΤΕΡΑ

Στα κυκλώματα του υπολογιστή τα δεδομένα αποθηκεύονται με δυαδική μορφή, δηλαδή 0 και 1, ανεξάρτητα από το πως τα ορίζει ο προγραμματιστής, όπως ακεραίους ή πραγματικούς σε δεκαδικό σύστημα, ή ακόμη χαρακτήρες κ.λπ. Έτσι ο αριθμός 17 του δεκαδικού συστήματος ισοδυναμεί με τον αριθμό **00010001** του δυαδικού συστήματος, ο οποίος μπορεί να αποθηκευτεί σε ένα byte. Αν μετακινήσουμε τα ψηφία αυτά κατά μία θέση προς τα αριστερά, δηλαδή αν προσθέσουμε ένα 0 στο τέλος του αριθμού και αγνοήσουμε το αρχικό 0, τότε προκύπτει ο αριθμός **00100010** του δυαδικού συστήματος, που ισοδυναμεί με το αριθμό 34 του δεκαδικού συστήματος. Επίσης, με παρόμοιο τρόπο, αν μετακινήσουμε τα ψηφία κατά μία θέση δεξιά, δηλαδή αποκόψουμε το τελευταίο ψηφίο 1 και θεωρήσουμε ένα ακόμη αρχικό 0, τότε προκύπτει ο αριθμός **00001000** του δυαδικού συστήματος, που ισοδυναμεί με τον αριθμό 8 του δεκαδικού συστήματος. Άρα η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί δύο, ενώ η ολίσθηση προς τα δεξιά ισοδυναμεί με την ακέραια διαίρεση διά δύο.

Η μέθοδος αυτή χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ότι ο γνωστός μας χειρωνακτικός τρόπος πολλαπλασιασμού. Πιο συγκεκριμένα, απαιτεί πολλαπλασιασμό επί δύο, διαίρεση διά δύο και πρόσθεση. Σε αντίθεση η γνωστή μας διαδικασία πολλαπλασιασμού απαιτεί πολλαπλασιασμό με οποιοδήποτε ακέραιο και πρόσθεση.

Σε επίπεδο, λοιπόν, κυκλωμάτων υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση διά δύο μπορούν να υλοποιηθούν ταχύτατα με μία απλή εντολή ολίσθησης (**SHIFT**), σε αντίθεση με τον πολλαπλασιασμό με οποιοδήποτε ακέραιο που θεωρείται πιο χρονοβόρα διαδικασία. Το τελευταίο γεγονός είναι ο λόγος που ο πολλαπλασιασμός **ΑΛΛΑ ΡΩΣΙΚΑ** είναι προτιμότερος απ' ότι ο χειρωνακτικός τρόπος πολλαπλασιασμού δύο ακεραίων.

Ακολουθεί παράδειγμα:

Α) Έστω ο δυαδικός αριθμός **10011001**. Θα τον ολισθήσουμε **ΔΕΞΙΑ**.

128	64	32	16	8	4	2	1		
1	0	0	1	1	0	0	1		153
0	1	0	0	1	1	0	0	1	76
0	0	1	0	0	1	1	0	0	38
0	0	0	1	0	0	1	1	0	19
0	0	0	0	1	0	0	1	1	9
0	0	0	0	0	1	0	0	0	4
0	0	0	0	0	0	1	0	0	2
0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0	1	0

ΑΡΑ ΕΝΑΣ ΔΥΑΔΙΚΟΣ ΑΡΙΘΜΟΣ ΟΤΑΝ ΟΛΙΣΘΑΙΝΕΙ ΚΑΤΑ ΜΙΑ ΘΕΣΗ ΔΕΞΙΑ ΤΟΤΕ :
ΔΙΑΙΡΕΙΤΑΙ ΑΚΕΡΑΙΑ ΜΕ ΤΟ 2

Β) Έστω ο δυαδικός αριθμός **11001**. Θα τον ολισθήσουμε **ΑΡΙΣΤΕΡΑ**.

128	64	32	16	8	4	2	1	
0	0	0	1	1	0	0	1	25
0	0	1	1	0	0	1	0	50
0	1	1	0	0	1	0	0	100
1	1	0	0	1	0	0	0	200

Και συνεχίζουμε μέχρι όπου χρειαστεί.

**ΑΡΑ ΕΝΑΣ ΔΥΑΔΙΚΟΣ ΑΡΙΘΜΟΣ ΟΤΑΝ ΟΛΙΣΘΕΝΕΙ ΚΑΤΑ ΜΙΑ ΘΕΣΗ ΑΡΙΣΤΕΡΑ ΤΟΤΕ:
ΠΟΛΛΑΠΛΑΣΙΑΖΕΤΑΙ ΜΕ ΤΟ 2**

2.6 ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ ΑΛΛΑ ΡΩΣΙΚΑ

Στη συνέχεια προχωρούμε στην ανάπτυξη ενός συνθετότερου προβλήματος, όπου για την αλγοριθμική του επίλυση γίνεται χρήση αρκετών από τις προηγούμενες δομές. Ας θεωρήσουμε την πράξη του πολλαπλασιασμού δύο ακεραίων αριθμών και ας θυμηθούμε πως αυτή υλοποιείται χειρωνακτικά. Τοποθετούμε, λοιπόν, τους δύο αριθμούς τον ένα κάτω από τον άλλο και πολλαπλασιάζουμε κάθε ψηφίο του κάτω αριθμού με όλα ψηφία του επάνω αριθμού. Πιο συγκεκριμένα, για κάθε ψηφίο του κάτω αριθμού παράγεται ένα μερικό γινόμενο, ενώ τα μερικά γινόμενα τοποθετούνται το ένα κάτω από το άλλο με μία μετατόπιση από τα δεξιά προς τα αριστερά καθώς θεωρούμε διαδοχικά τα ψηφία των μονάδων, των δεκάδων, των εκατοντάδων κ.λπ.

Ωστόσο, η πράξη του πολλαπλασιασμού δεν εκτελείται από τον υπολογιστή με τον τρόπο αυτό. Πιο συγκεκριμένα, ο χρησιμοποιούμενος τρόπος είναι ο λεγόμενος πολλαπλασιασμός **ΑΛΛΑ ΡΩΣΙΚΑ**.

Χωρίς βλάβη της γενικότητας θεωρούμε ότι οι ακέραιοι είναι θετικοί (μεγαλύτεροι του μηδενός), αλλά η μέθοδος μπορεί εύκολα να μετατραπεί, ώστε να περιγράψει και την περίπτωση των αρνητικών ακεραίων. Πώς ακριβώς λειτουργεί η μέθοδος, θα φανεί με το επόμενο παράδειγμα, όπου περιγράφεται ο αλγόριθμος με ελεύθερο κείμενο.

Έστω, λοιπόν, ότι δίνονται δύο θετικοί ακέραιοι αριθμοί, οι αριθμοί 8 και 1486. Οι αριθμοί γράφονται δίπλα-δίπλα και ο πρώτος διπλασιάζεται, ενώ ο δεύτερος υποδιπλασιάζεται αγνοώντας το δεκαδικό μέρος. Στο **Σχήμα Α1** και **ΣΧΗΜΑ Α2** παρουσιάζεται η επαναλαμβανόμενη διαδικασία, που συνεχίζεται **Μέχρις_ότου** στη δεύτερη στήλη να προκύψει μονάδα ή **Μέχρις_ότου** στη δεύτερη στήλη προκύψει μηδέν. Τελικώς, το γινόμενο ισούται με το άθροισμα των στοιχείων της πρώτης στήλης, όπου αντίστοιχα στη δεύτερη στήλη υπάρχει περιττός αριθμός.

Για το παράδειγμά μας, τα στοιχεία αυτά παρουσιάζονται στην τρίτη στήλη.

1) Θετικός ? 2) Μονός ? 3β) Χ 2 3α) DIV 2		
ΑΡΙΘΜΟΣ_1	ΑΡΙΘΜΟΣ_2	ΑΠΟΤΕΛΕΣΜΑ
8	1486	0
16	743	0+16=16
32	371	16+32=48
64	185	48+64=112
128	92	
256	46	
512	23	112+512=624
1024	11	624+1024=1648
2048	5	1648+2048=3696
4096	2	
8192	1	3696+8192=11888
16384	0	

ΣΧΗΜΑ Α1

1) Θετικός ? 2) Μονός ? 3β) Χ 2 3α) DIV 2		
ΑΡΙΘΜΟΣ_1	ΑΡΙΘΜΟΣ_2	ΑΠΟΤΕΛΕΣΜΑ
1486	8	0
2972	4	
5944	2	
11888	1	0+11888=11888
23766	0	

ΣΧΗΜΑ Α2

Στη συνέχεια παρουσιάζεται ο αλγόριθμος πολλαπλασιασμού ακεραίων αλλά ρωσικά με φυσική γλώσσα κατά βήματα.

Αλγόριθμος: Πολλαπλασιασμός δύο θετικών ακεραίων (ΑΛΑ ΡΩΣΙΚΑ)

Είσοδος: Δύο ακέραιοι Αριθμός_1 και Αριθμός_2, όπου Αριθμός_2>0 (ομοίως Αριθμός_2 > 1)

Έξοδος: Το γινόμενο Αποτέλεσμα = Αριθμός_1* Αριθμός_2

Βήμα 1 Θέσε Αποτέλεσμα=0

Βήμα 2 Αν Αριθμός_2>0, τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7

Βήμα 3 Αν ο Αριθμός 2 είναι περιττός, τότε θέσε Αποτέλεσμα = Αποτέλεσμα + Αριθμός_1

Βήμα 4 Θέσε Αριθμός_1= Αριθμός_1*2

Βήμα 5 Θέσε Αριθμός_2= Αριθμός_2/2 (θεώρησε μόνο το ακέραιο μέρος)

Βήμα 6 Πήγαινε στο Βήμα 2

Βήμα 7 Τύπωσε το Αποτέλεσμα.

Ακολουθεί ο αλγόριθμος σε ψευδοκώδικα για το ίδιο πρόβλημα του πολλαπλασιασμού αλλά ρωσικά για θετικούς αριθμούς.

```

Αλγόριθμος Αλά_Ρωσικά
    Αποτέλεσμα  $\leftarrow$  0
    Διάβασε Αριθμός_1 , Αριθμός_2
    Όσο Αριθμός_2 > 0 επανάλαβε
        Αν Αριθμός_2 mod 2 = 1 τότε
            Αποτέλεσμα  $\leftarrow$  Αποτέλεσμα + Αριθμός_1
        Τέλος_αν
        Αριθμός_2  $\leftarrow$  Αριθμός_2 div 2
        !ή Αριθμός_2  $\leftarrow$  Αριθμός_2 / 2
        Αριθμός_1  $\leftarrow$  Αριθμός_1 * 2
    Τέλος_επανάληψης
    Εμφάνισε Αποτέλεσμα
Τέλος Αλά_Ρωσικά
    
```

Ακολουθεί ο αλγόριθμος σε ψευδοκώδικα για το ίδιο πρόβλημα του πολλαπλασιασμού αλλά ρωσικά για αρνητικούς αριθμούς.

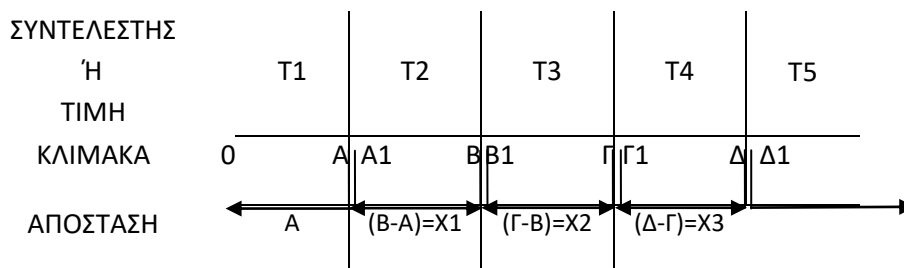
```

Αλγόριθμος Αλά_Ρωσικά
    Αποτέλεσμα  $\leftarrow$  0
    Διάβασε Αριθμός_1 , Αριθμός_2
    Θετικός_1  $\leftarrow$  Αληθής
    Θετικός_2  $\leftarrow$  Αληθής
    Αν Αριθμός_1 < 0 τότε
        Θετικός_1  $\leftarrow$  Ψευδής
        Αριθμός_1  $\leftarrow$  Αριθμός_1 * (-1)
    Τέλος_αν
    Αν Αριθμός_2 < 0 τότε
        Θετικός_2  $\leftarrow$  Ψευδής
        Αριθμός_2  $\leftarrow$  Αριθμός_2 * (-1)
    Τέλος_αν
    Όσο Αριθμός_2 > 0 επανάλαβε
        Αν Αριθμός_2 mod 2 = 1 τότε
            Αποτέλεσμα  $\leftarrow$  Αποτέλεσμα + Αριθμός_1
        Τέλος_αν
        Αριθμός_2  $\leftarrow$  Αριθμός_2 div 2
        !ή Αριθμός_2  $\leftarrow$  Αριθμός_2 / 2 για πραγματικούς αριθμούς
        Αριθμός_1  $\leftarrow$  Αριθμός_1 * 2
    Τέλος_επανάληψης
    Αν Θετικός_1 = Ψευδής τότε
        Αποτέλεσμα  $\leftarrow$  Αποτέλεσμα * (-1)
    Τέλος_αν
    Αν Θετικός_2 = Ψευδής τότε
        Αποτέλεσμα  $\leftarrow$  Αποτέλεσμα * (-1)
    Τέλος_αν
    Εμφάνισε Αποτέλεσμα
Τέλος Αλά_Ρωσικά
    
```

ΜΕΘΟΔΟΛΟΓΙΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕ ΚΛΙΜΑΚΩΤΟ ΥΠΟΛΟΓΙΣΜΟ

Έστω ότι η άσκηση αναφέρει την παρακάτω γενική κλίμακα για τον υπολογισμό Έκπτωσης ή Ποσό Πληρωμής με βάση Ποσό Αγοράς ή Ποσότητα αγοράς αντίστοιχα :

Ποσότητα ή Ποσό Αγοράς (σε €)	Έκπτωση % ή ΤΙΜΗ (σε €)
Μέχρι Α	T1
A1 – Β	T2
B1 – Γ	T3
Γ1 – Δ	T4
Δ1 και άνω	T5

1) Σχεδιάζουμε άξονα τιμών:

2) Γράφουμε τον αλγόριθμο του υπολογισμού του ζητούμενου

```

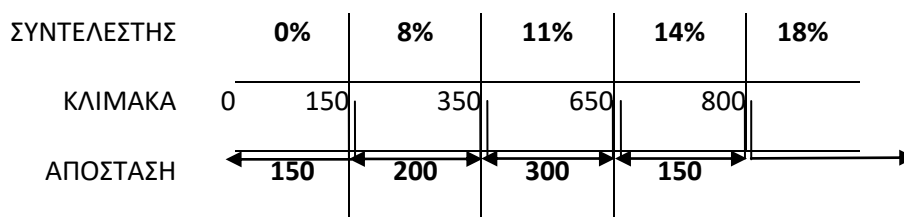
ΔΙΑΒΑΣΕ ΤΙΜΗ
ΑΝ ΤΙΜΗ <=Α ΤΟΤΕ
    ΖΗΤ_ΤΙΜΗ ← ΤΙΜΗ *Τ1
ΑΛΛΙΩΣ_ΑΝ ΤΙΜΗ <=Β ΤΟΤΕ
    ΖΗΤ_ΤΙΜΗ ← Α*Τ1+(ΤΙΜΗ -Α)*Τ2
ΑΛΛΙΩΣ_ΑΝ ΤΙΜΗ <=Γ ΤΟΤΕ
    ΖΗΤ_ΤΙΜΗ ← Α*Τ1+Χ1*Τ2+(ΤΙΜΗ -Β)*Τ3
ΑΛΛΙΩΣ_ΑΝ ΤΙΜΗ <=Δ ΤΟΤΕ
    ΖΗΤ_ΤΙΜΗ ← Α*Τ1+Χ1*Τ2+Χ2*Τ3+(ΤΙΜΗ -Γ)*Τ4
ΑΛΛΙΩΣ
    ΖΗΤ_ΤΙΜΗ ← Α*Τ1+Χ1*Τ2+Χ2*Τ3+Χ3*Τ4+( ΤΙΜΗ -Δ)*Τ5
ΤΕΛΟΣ_ΑΝ
ΕΜΦΑΝΙΣΕ ΖΗΤ_ΤΙΜΗ
    
```

ΠΑΡΑΔΕΙΓΜΑΤΑ ΓΙΑ ΚΑΤΑΝΟΗΣΗ

Α) Έστω ότι η άσκηση αναφέρει την παρακάτω κλίμακα για τον υπολογισμό της έκπτωσης με βάση το ποσό αγορών και τον υπολογισμό του ποσού πληρωμής:

Ποσό (σε €)	Έκπτωση %
Μέχρι 150	0
151 – 350	8
351 – 650	11
651 – 800	14
801 και άνω	18

1. Σχεδιάζουμε άξονα τιμών:



2. Γράφουμε τον αλγόριθμο του υπολογισμού της έκπτωσης που αναλογεί.

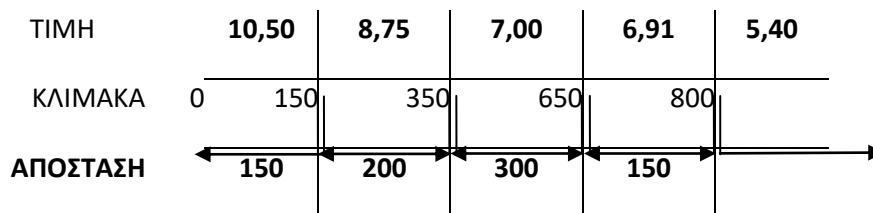
```

ΔΙΑΒΑΣΕ ΠΟΣΟ
ΑΝ ΠΟΣΟ<=150 ΤΟΤΕ
    ΕΚΠΤΩΣΗ←ΠΟΣΟ*0/100
ΑΛΛΙΩΣ_ΑΝ ΠΟΣΟ<=350 ΤΟΤΕ
    ΕΚΠΤΩΣΗ←150*0/100+(ΠΟΣΟ-150)*8/100
ΑΛΛΙΩΣ_ΑΝ ΠΟΣΟ<=650 ΤΟΤΕ
    ΕΚΠΤΩΣΗ←150*0/100+200*8/100+(ΠΟΣΟ-350)*11/100
ΑΛΛΙΩΣ_ΑΝ ΠΟΣΟ<=800 ΤΟΤΕ
    ΕΚΠΤΩΣΗ←150*0/100+200*8/100+300*11/100+(ΠΟΣΟ-650)*14/100
ΑΛΛΙΩΣ
    ΕΚΠΤΩΣΗ←150*0/100+200*8/100+300*11/100+150*14/100+(ΠΟΣΟ-800)*18/100
ΤΕΛΟΣ_ΑΝ
ΠΟΣΟ_ΠΛ←ΠΟΣΟ-ΕΚΠΤΩΣΗ
ΕΜΦΑΝΙΣΕ ΠΟΣΟ_ΠΛ
    
```

Β) Έστω ότι η άσκηση αναφέρει την παρακάτω κλίμακα για τον υπολογισμό του ποσού Πληρωμής με βάση το ποσότητα αγορών :

Ποσότητα	ΤΙΜΗ (σε €)
Μέχρι 150	10,50
151 – 350	8,75
351 – 650	7,00
651 – 800	6,91
801 και άνω	5,40

1) Σχεδιάζουμε άξονα τιμών:



2) Γράφουμε τον αλγόριθμο του κλιμακωτού υπολογισμού για τον υπολογισμό του ποσού πληρωμής που αναλογεί.

ΔΙΑΒΑΣΕ ΠΟΣΟΤΗΤΑ

ΑΝ ΠΟΣΟΤΗΤΑ ≤ 150 ΤΟΤΕ

ΠΟΣΟ_ΠΛ ← ΠΟΣΟ * 10,50

ΑΛΛΙΩΣ_ΑΝ ΠΟΣΟΤΗΤΑ ≤ 350 ΤΟΤΕ

ΠΟΣΟ_ΠΛ ← 150 * 10,50 + (ΠΟΣΟΤΗΤΑ - 150) * 8,75

ΑΛΛΙΩΣ_ΑΝ ΠΟΣΟΤΗΤΑ ≤ 650 ΤΟΤΕ

ΠΟΣΟ_ΠΛ ← 150 * 10,50 + 200 * 8,75 + (ΠΟΣΟΤΗΤΑ - 350) * 7,00

ΑΛΛΙΩΣ_ΑΝ ΠΟΣΟΤΗΤΑ ≤ 800 ΤΟΤΕ

ΠΟΣΟ_ΠΛ ← 150 * 10,50 + 200 * 8,75 + 300 * 7,00 + (ΠΟΣΟΤΗΤΑ - 650) * 6,91

ΑΛΛΙΩΣ

ΠΟΣΟ_ΠΛ ← 150 * 10,50 + 200 * 8,75 + 300 * 7,00 + 150 * 6,91 + (ΠΟΣΟΤΗΤΑ - 800) * 5,40

ΤΕΛΟΣ_ΑΝ

ΕΜΦΑΝΙΣΕ ΠΟΣΟ_ΠΛ

ΕΡΩΤΗΣΕΙΣ ΘΕΩΡΙΑΣ

1. Δώστε τον ορισμό του αλγόριθμου.
2. Περιγράψτε τα κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος.
3. Με ποιούς τρόπους μπορεί να αναπαρασταθεί ένας αλγόριθμος; Ποιά μειονεκτήματα έχει ο καθένας;
4. Ποιά είναι τα σύμβολα που χρησιμοποιούμε στα διαγράμματα ροής; Σχεδιάστε και περιγράψτε τα.
5. Με ποιες εντολές ξεκινά και τερματίζεται ένας αλγόριθμος; Πώς μπορούμε να διαβάσουμε κάτι από το πληκτρολόγιο.
6. Πώς μπορούμε να γράψουμε κάτι στην οθόνη και στον εκτυπωτή;
7. Τι είναι οι σταθερές;
8. Τι είναι οι μεταβλητές; Ποια είναι τα είδη μεταβλητών που υπάρχουν;
9. Τι είναι τελεστές; Ποια είδη υπάρχουν και ποιοι τελεστές ανά είδος;
10. Ποιες κατηγορίες τελεστών υπάρχουν; Αναφέρετε όλους τους τελεστές ανά κατηγορία.
11. Τι είναι οι εκφράσεις; Τι κάνει η εντολή εκχώρησης τιμής;
12. Γράψτε τις τρεις διαφορετικές μορφές της εντολής ΑΝ.
13. Με ποιους δύο τρόπους μπορούμε να λάβουμε πολλές διαφορετικές αποφάσεις σε έναν αλγόριθμο;
14. Περιγράψτε τους λογικούς τελεστές **Ή**, **ΚΑΙ**, **ΌΧΙ**.
15. Δώστε ένα παράδειγμα της εντολής **ΟΣΟ**, **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΓΙΑ**.

ΕΡΩΤΗΣΕΙΣ ΣΩΣΤΟΥ – ΛΑΘΟΥΣ

1. Ο αλγόριθμος είναι απαραίτητος μόνο για την επίλυση προβλημάτων πληροφορικής.
2. Ο αλγόριθμος αποτελείται από ένα πεπερασμένο σύνολο εντολών.
3. Ο αλγόριθμος μπορεί να περιλαμβάνει και εντολές που δεν είναι σαφείς.
4. Η αναπαράσταση των αλγορίθμων μπορεί να γίνει με χρήση ελεύθερου κειμένου και φυσικής γλώσσας.
5. Τα κυριότερα σύμβολα των διαγραμμάτων ροής είναι η έλλειψη, ο ρόμβος, το ορθογώνιο και το πλάγιο παραλληλόγραμμο.
6. Η δομή της ακολουθίας είναι ιδιαίτερα χρήσιμη για την αντιμετώπιση πολύπλοκων προβλημάτων.
7. Μια σταθερά μπορεί να αλλάξει τιμή κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
8. Μια μεταβλητή μπορεί να αλλάζει τιμή και όνομα κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
9. Μια μεταβλητή μπορεί να αλλάζει τύπο δεδομένων κατά τη διάρκεια εκτέλεσης ενός αλγορίθμου.
10. Μια μεταβλητή μπορεί να υποθηκεύσει και αφαιρετικά δεδομένα.
11. Δεξιά μιας εντολής εκχώρησης τιμής δεν μπορεί να βρίσκεται η ίδια μεταβλητή που βρίσκεται αριστερά.
12. Όλα τα προβλήματα λύνονται και αλγοριθμικά.
13. Τα σχόλια διευκολύνουν την κατανόηση ενός αλγορίθμου.
14. Σε μια εντολή εκχώρησης δεν επιτρέπεται η χρήση σταθερών.
15. Ένας αλγόριθμος επιλύει μόνο υπολογιστικά προβλήματα.
16. Για την αναπαράσταση των δεδομένων εισόδου ενός αλγορίθμου χρησιμοποιούμε τις σταθερές.

17. Η σειρά εκτέλεσης των εντολών στη δομή ακολουθίας είναι προκαθορισμένη.
18. Στη δομή ακολουθίας εκτελούνται όλες οι εντολές.
19. Στο δεξί τμήμα μιας εντολής εκχώρησης πρέπει να υπάρχει υποχρεωτικά πράξη.
20. Τα είδη των μεταβλητών που χρησιμοποιούμε είναι οι αριθμητικές, οι αλφαριθμητικές και οι σταθερές.
21. Για να αναπαραστήσουμε τα δεδομένα και τα αποτελέσματα σ' έναν αλγόριθμο, χρησιμοποιούμε σταθερές.
22. Στο διάγραμμα ροής το σχήμα του ρόμβου δηλώνει το τέλος ενός αλγορίθμου.
23. Η εντολή εκχώρησης τιμής αποδίδει το αποτέλεσμα μιας έκφρασης (παράστασης) σε μια μεταβλητή.
24. Σε μια εντολή εκχώρησης είναι δυνατόν μια παράσταση στο δεξιό μέλος να περιέχει τη μεταβλητή που βρίσκεται στο αριστερό.
25. Στη δομή απλής επιλογής η ομάδα εντολών εντός της δομής εκτελείται όταν η συνθήκη είναι αληθής.
26. Στην πολλαπλή επιλογή κάθε περίπτωση αντιστοιχεί σε διαφορετική τιμή της συνθήκης.
27. Για τον υπολογισμό του μέσου όρου αριθμών πρέπει να χρησιμοποιηθεί η δομή επιλογής.
28. Στη δομή επιλογής υπάρχει περίπτωση κάποιες εντολές να μην εκτελεστούν ποτέ.
29. Κάθε εντολή Αν περιέχει Αλλιώς.
30. Κάθε εντολή πολλαπλής επιλογής μπορεί να αναπαρασταθεί από πολλά απλά Αν.
31. Στην δομή Επίλεξε εκτελείται πάντα το "Περίπτωση Αλλιώς".
32. Όταν πρέπει να εκτελεστούν κάποιες εντολές υπό κάποια συνθήκη χρησιμοποιείται η δομή ακολουθίας.
33. Μια δομή επιλογής μπορεί να περιλαμβάνει μόνο εντολές εκχώρησης τιμής.
34. Σε μια έκφραση εκτελούνται πρώτα οι συγκριτικοί τελεστές και στη συνέχεια οι αριθμητικοί.
35. Στη δομή επανάληψης Για δεν είναι δυνατόν η αρχική τιμή να είναι να είναι μεγαλύτερη από την τελική.
36. Δεν μπορούμε να έχουμε μια δομή επανάληψης μέσα σε μια άλλη δομή επανάληψης.
37. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό.
38. Με χρήση της εντολής "Όσο...επανάλαβε" επιτυγχάνεται η επανάληψη μίας διαδικασίας με βάση κάποια συνθήκη.
39. Με την εντολή "Αρχή_επανάληψης...Μέχρις_ότου..." υπάρχει ένας βρόχος που εκτελείται τουλάχιστον μία φορά.
40. Η εντολή "Για i από .. μέχρι .. βήμα .." πρέπει να περιλαμβάνει για βήμα πάντοτε ένα θετικό αριθμό.
41. Στη δομή επανάληψης Για το βήμα δεν μπορεί να είναι μηδέν.
42. Όταν το πλήθος των επαναλήψεων είναι γνωστό δεν μπορεί να χρησιμοποιηθεί η δομή επανάληψης "Όσο...επανάλαβε".
43. Οι εντολές του βρόχου "Για" εκτελούνται τουλάχιστον μια φορά.
44. Στην δομή «Όσο», η ομάδα εντολών εκτελείται μέχρι η συνθήκη να γίνει ψευδής.
45. Στην δομή «Μέχρις_ότου», υπάρχει περίπτωση η ομάδα εντολών του βρόχου να μην εκτελεστεί καμία φορά.
46. Στις δομές "Όσο" και "Μέχρις_ότου", οι συνθήκες είναι μεταξύ τους αντίθετες.
47. Η δομή "Όσο...Επανάλαβε" χρησιμοποιείται μόνο όταν γνωρίζουμε το πλήθος των επαναλήψεων.
48. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση

της δομής επανάληψης "Για".

49. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Όσο".
50. Κάθε πρόβλημα που απαιτεί τη χρήση δομής επανάληψης μπορεί να επιλυθεί με τη χρήση της δομής επανάληψης "Μέχρις_ότου".
51. Η δομή "Μέχρις_ότου" τερματίζεται όταν η συνθήκη είναι αληθής.
52. Κάθε βρόχος "Για" μπορεί να μετατραπεί σε "Όσο".
53. Κάθε βρόχος "Όσο" μπορεί να μετατραπεί σε "Για".
54. Κάθε βρόχος "Όσο" μπορεί να μετατραπεί σε "Μέχρις_ότου".
55. Εντός μιας δομής επιλογής δεν μπορεί να περιέχεται δομή επανάληψης.
56. Εντός μιας δομής επανάληψης δεν μπορεί να περιέχεται δομή επιλογής.
57. Στην δομή επανάληψης "Μέχρις_ότου" οι μεταβλητές που συμμετέχουν στην συνθήκη πρέπει να αρχικοποιούνται πριν το βρόχο.
58. Στην δομή επανάληψης "Όσο" οι μεταβλητές που συμμετέχουν στην συνθήκη πρέπει να πάρουν τιμή πριν το βρόχο.
59. Οι επαναληπτικές δομές χρησιμοποιούνται στην περίπτωση που μια ομάδα εντολών πρέπει να εκτελεστεί πολλές φορές.
60. Μια δομή επανάληψης πρέπει να φροντίζει για μεταβολή της τιμής της συνθήκης ώστε κάποτε να τερματίζεται.
61. Στη δομή επανάληψης Για πρέπει η τιμή του μετρητή να μεταβάλλεται εντός του βρόχου.
62. Το αποτέλεσμα μια πράξης μπορεί να εκχωρηθεί σε μια σταθερά.
63. Δεσμευμένες λέξεις ονομάζονται αυτές που ορίζει ο προγραμματιστής ως ονομασίες των μεταβλητών που χρησιμοποιεί.
64. Η εντολή $X \leftarrow X * X$ είναι έγκυρη.
65. Στη δομή ακολουθίας μια συγκεκριμένη εντολή μπορεί να εκτελεστεί πολλές φορές.
66. Η είσοδος σε ένα αλγοριθμικό πρόβλημα είναι ένα σύνολο μεταβλητών που σχετίζονται με τα δεδομένα του.
67. Χρησιμοποιούμε τη δομή επιλογής όταν θέλουμε μια ομάδα εντολών να εκτελεστεί πολλές φορές
68. Η σύζευξη δύο λογικών συνθηκών είναι ψευδής όταν μόνο μία από τις δύο λογικές συνθήκες είναι αληθής
69. Μια δομή επιλογής μπορεί να εκτελεστεί πολλές φορές
70. Η δομή της επιλογής χρησιμοποιείται στις περιπτώσεις όπου υπάρχει μία συγκεκριμένη σειρά βημάτων για την επίλυση ενός προβλήματος
71. Όταν χρειάζεται να υπάρξει απόφαση με βάση κάποιο κριτήριο, τότε χρησιμοποιείται η δομή της επιλογής.
72. Η δομή της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής).
73. Οι διαδικασίες των πολλαπλών επιλογών εφαρμόζονται στα προβλήματα όπου εκτελούνται κάποιες εντολές ανάλογα με την τιμή που παίρνει μία μεταβλητή
74. Μία εντολή «Αν...τότε» δεν μπορεί να περιληφθεί στα όρια κάποιας άλλης εντολής "Αν...τότε"
75. Με την εντολή "Αν $X \text{ div } 2 = 0$ " ελέγχουμε αν ο X είναι άρτιος
76. Η λογική πρόταση " $X^2 \geq 0$ " είναι πάντοτε αληθής
77. Όταν σε μια δομή "Για" παραλείπεται το βήμα, τότε εννοείται ως βήμα το 1
78. Η δομή "Όσο" τερματίζεται όταν η συνθήκη γίνεται ψευδής
79. Εντός της δομής "Για" δεν επιτρέπεται η τροποποίηση της τιμής του μετρητή

Ερωτήσεις ΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

- Πόσο κάνει η παρακάτω πράξη: $5 \bmod 2 * 10$.
α. 10 β. 5 γ. 0 δ. απροσδιόριστο
- Οι εμφωλευμένες δομές περιλαμβάνουν συνδυασμό:
α) συνθήκης και εκτύπωσης β) διαφόρων αλγοριθμικών δομών
γ) συνθήκης και ανάγνωσης δ) ανάγνωσης και εκτύπωσης
- Με την δομή "Όσο (($a \bmod 2 = 0$) και ($b \bmod 2 = 1$)) επανάλαβε", πετυχαίνουμε να εκτελούμε τον βρόχο όσο
α) ο a είναι περιττός και ο b άρτιος β) ο a είναι άρτιος και ο b περιττός
γ) ο a και ο b είναι άρτιοι δ) ο a και ο b είναι περιττοί
- Η παράσταση: $(3(a^2 - 4b^2) - 5(a^2c + b^2d)) / (abc - d^2)$ σε ποιά από τις παρακάτω εκχωρήσεις τιμών αντιστοιχεί;
α) $f \leftarrow 3*(a*a-4*b*b)-5*(a*a*c+b*b*d)/(a*b*c-d*d)$
β) $f \leftarrow 3*(a*a-4*b*b)-5*(a*a*c+b*b*d)/a*b*c-d*d$
γ) $f \leftarrow (3*(a*a-4*b*b)-5*(a*a*c+b*b*d))/(a*b*c-d*d)$
δ) $f \leftarrow (3*(a*a-4*b*b))-(5*(a*a*c+b*b*d))/(a*b*c-d*d)$
- Αν οι μεταβλητές a, b είναι αριθμητικές και έχουν κάποια τιμή, τότε οι παρακάτω εντολές ανταλλάσσουν τις τιμές τους.
 $A \leftarrow A + B$
 $B \leftarrow A - B$
 $A \leftarrow A - B$
- Μία εντολή «Αν...τότε» περιλαμβάνει κάποια:
α) συνθήκη β) ακολουθία γ) ανάθεση δ) επανάληψη
- Με την ερώτηση "Αν ($A \bmod 2 = 0$)", εννοούμε εν γένει, ότι επιθυμούμε να εξετάσουμε αν
α) ο A είναι περιττός β) ο A είναι μικρότερος του 2
γ) ο A ισούται με 2 δ) ο A διαιρείται ακριβώς με το 2
- Η εκτέλεση του κάτωθι τμήματος αλγορίθμου:
 $s \leftarrow 0$
 $p \leftarrow 0$
Διάβασε a
Όσο ($a > 0$) επανάλαβε
 Αν ($a \bmod 2 = 1$) τότε
 $s \leftarrow s + a$
 αλλιώς
 $p \leftarrow p * a$
Τέλος_αν
Διάβασε a
Τέλος_επανάληψης
όπου a ακέραιος, μας δίνει
α) το γινόμενο των περιττών και 0 για τους αρτίους
β) το άθροισμα των περιττών και το γινόμενο των άρτιων
γ) το άθροισμα των περιττών και 0 για τους άρτιους
δ) το άθροισμα των αρτίων και 0 για τους περιττούς

9. Δίνεται το παρακάτω τμήμα αλγορίθμου

```

da ← 0
dp ← 0
Για i από a μέχρι 4 με_βήμα -2
    Αν (i mod 2=0) τότε
        da ← da + 1
    αλλιώς
        dp ← dp + 1
Τέλος_αν
Τέλος_επανάληψης

```

αν το αποτέλεσμα είναι da=0 και dp=3, τότε ποια τιμή θα μπορούσε να έχει το a;

α) a=11 β) a=9 γ) a=8 δ) a=2

10. Κάθε αλγόριθμος πρέπει να ικανοποιεί το κριτήριο της:

α) επιλογής	γ) ανάθεσης
β) ακολουθίας	δ) περατότητας

11. Η επιστήμη της Πληροφορικής περιλαμβάνει τη μελέτη των αλγορίθμων μεταξύ άλλων και από τη σκοπιά:

α) υλικού	γ) αποτελεσματικότητας
β) ελεύθερου κειμένου	δ) ανάγνωσης / εκτύπωσης

12. Ένας από τους τρόπους αναπαράστασης των αλγορίθμων είναι:

α) λογικές εκφράσεις	γ) διαγραμματικές τεχνικές
β) θεωρητική τυποποίηση	δ) αριθμητικές πράξεις

13. Ποιά από τα παρακάτω αποτελούν εντολές της ψευδογλώσσας των αλγορίθμων:

α) $A + B = 1$	γ) $A + B \leftarrow 23$
β) $A \leftarrow B * 5$	δ) $A \leftarrow 2 * B \leftarrow 12$

14. Μία εμφωλευμένη δομή χρησιμοποιείται όταν χρειάζεται:

- α) μία ενέργεια να περιληφθεί μέσα σε άλλη ενέργεια
- β) να υπάρχει επανάληψη τυποποιημένων ενεργειών
- γ) να υπάρχει εκτύπωση και ανάγνωση τιμών
- δ) να επαναληφθεί μία ενέργεια πολλές φορές

15. Η λογική πράξη (H) μεταξύ 2 προτάσεων είναι αληθής όταν:

- α) οποιαδήποτε από τις δύο προτάσεις είναι αληθής
- β) η πρώτη πρόταση είναι ψευδής
- γ) η δεύτερη πρόταση είναι ψευδής
- δ) και οι δύο προτάσεις είναι αληθής

16. Η λογική πράξη (ΚΑΙ) μεταξύ 2 προτάσεων είναι αληθής όταν:

- α) οποιαδήποτε από τις δύο προτάσεις είναι αληθής
- β) η πρώτη πρόταση είναι αληθής
- γ) η δεύτερη πρόταση είναι αληθής
- δ) και οι δύο προτάσεις είναι αληθείς

17. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις όπου:

- I. μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε δύο περιπτώσεις
- II. μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων
- III. υπάρχει απαίτηση να ληφθεί μία απόφαση με βάση κάποια συνθήκη
- IV. υπάρχουν δύο συνθήκες που πρέπει να ισχύουν η μία μετά την άλλη

18. Αν μετά την εκτέλεση του κάτωθι τμήματος αλγορίθμου:

Αν $(x \bmod y < x \div y)$ **τότε**

$a \leftarrow 0$

$b \leftarrow 0$

Αλλιώς

$a \leftarrow x \div y$

$b \leftarrow x \bmod y$

Τέλος_αν

το $a = 0$ και το $b = 3$, τι τιμές θα μπορούσαν να έχουν τα x και y ;

- α) $x=7, y=2$ β) $x=4, y=3$ γ) $x=3, y=5$ δ) $x=9, y=3$

19. Τι θα εκτυπωθεί, μετά την εκτέλεση του παρακάτω αλγορίθμου:

$\alpha \leftarrow 2$

$\beta \leftarrow -3$

Όσο $\beta \leq 0$ **επανάλαβε**

$\beta \leftarrow \beta + 1$

$\alpha \leftarrow \alpha + \beta - 1$

Τέλος_επανάληψης

Εκτύπωσε α

- α) 2 β) -4 γ) -1 δ) 4

20. Οι εντολές μεταξύ του "Αρχή_Επανάληψης...Μέχρις_ότου" ...

- α. εκτελούνται μέχρι η συνθήκη να γίνει αληθής
- β. εκτελούνται μέχρι η συνθήκη να γίνει ψευδής
- γ. μπορεί να μην εκτελεστούν καμία επανάληψη
- δ. θα εκτελεστούν οπωσδήποτε μια φορά

21. Το παρακάτω τμήμα αλγορίθμου

$\Sigma \leftarrow 0$

Για i **από** 100 **μέχρι** 999 **με_βήμα** 2

$\Sigma \leftarrow \Sigma + i$

Τέλος_επανάληψης

- α. Υπολογίζει το άθροισμα των τριψήφων αριθμών
- β. Υπολογίζει το άθροισμα των τριψήφων άρτιων αριθμών
- γ. Υπολογίζει το άθροισμα των τριψήφων περιττών

22. Μία διαδικασία που δεν ολοκληρώνεται μετά από πεπερασμένο πλήθος βημάτων δεν αποτελεί αλγόριθμο, αλλά:

- α) δεδομένα β) μία υπολογιστική διαδικασία
γ) μία εκτέλεση δ) ατέρμονα έλεγχο δεδομένων

ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΣΤΟΙΧΗΣΗΣ

1. Να συνδέσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β

Α Τιμή

1. "85"

2. 15

3. "Ψευδής"

4. Αληθής

5. "34.2"

6. -345.87

Β Τύπος Δεδομένων

Α) Αλφαριθμητικός (Συμβολοσειρά)

Β) Αριθμητικός (ακέραιος ή πραγματικός)

Γ) Λογικός

2. Να αντιστοιχίσετε τα στοιχεία των δυο στηλών

Α Εκφράσεις

1. $\alpha > \beta - 1$

2. Εκτύπωσε X

3. Αν $\alpha > 3$ τότε

...

Τέλος_αν

4. $\alpha \leftarrow \alpha + 2$

5. $\alpha + \beta / 2$

6. Διάβασε Σ

Β Αλγοριθμικές Έννοιες

Α) Δομή επιλογής

Β) Δομή επανάληψης

Γ) Αριθμητική έκφραση

Δ) Μεταβλητή

Ε) Εντολή εξόδου

ΣΤ) Εντολή εισόδου

Ζ) Λογική έκφραση

Η) Εντολή εκχώρησης τιμής

ΕΡΩΤΗΣΕΙΣ ΣΥΜΠΛΗΡΩΣΗΣ ΚΕΝΩΝ

- Οι _____ μεταβλητές μπορούν να λάβουν μόνο δυο τιμές: αληθής και ψευδής.
- Ο πολλαπλασιασμός _____ απαιτεί πολλαπλασιασμό επί δύο, διαίρεση δια δύο και πρόσθεση.
- Η επαναληπτική δομή που περιλαμβάνει έλεγχο επανάληψης στο τέλος της διαδικασίας ξεκινά με τη φράση «Αρχή_επανάληψης» και λήγει με τη φράση _____.
- Ο αλγόριθμος που δεν διαθέτει τρόπο τερματισμού χαρακτηρίζεται ως _____ βρόχος.
- Τα αναγνωριστικά των οποίων οι τιμές μεταβάλλονται κατά τη διάρκεια του αλγορίθμου ονομάζονται _____ και εκείνα των οποίων οι τιμές δεν μπορούν να μεταβληθούν _____.
- Η δομή "_____ i από τ1 μέχρι τ2 με_βήμα β" αποτελεί ένα επαναληπτικό σχήμα ορισμένων επαναλήψεων.
- Η επαναληπτική δομή "Όσο...Επανάλαβε" περιλαμβάνει διαδικασίες και λήγει με τη φράση _____.

ΆΛΥΤΕΣ ΑΣΚΗΣΕΙΣ

Α ΟΜΑΔΑ

1. Ποιά από τα κάτω αλφαριθμητικά είναι αποδεκτά ως ονόματα μεταβλητών σε έναν αλγόριθμο.

- i. Τιμή
- ii. Τιμή-1
- iii. Τιμή_2
- iv. Χασροτοποσ
- v. Τιμή.δ
- vi. τ

2. Να κρίνετε για την ορθότητά τους τις παρακάτω εντολές εκχώρησης τιμής.

- i. τιμή $\leftarrow$ "αρκετά"
- ii. τιμή $\leftarrow$ τιμή + 3
- iii. α + τιμή $\leftarrow$ 6
- iv. τιμή $\leftarrow$ ασ $\leftarrow$ 6
- v. τιμή $\leftarrow$ α*β+5
- vi. τιμή $\leftarrow$ "δ" + 5
- vii. τιμή $\leftarrow$ "τιμή" + 5

3. Πώς θα διατυπωθεί σε εντολή εκχώρησης τιμής, η κάθε μία από τις παρακάτω αλγεβρικές παραστάσεις.

A)

$$\frac{5x^3 + 7x^2 + 8}{8x - 6}$$

B)

$$6x^4 - z \left(\frac{7y+6}{2(x+3)} - 2 \right) + (9-y)^3$$

4. Ποιό είναι το αποτέλεσμα από την εκτέλεση των παρακάτω πράξεων.

- i. $14 \bmod 5 - 25 \bmod 8$
- ii. $3 * (3 \bmod 2) + 4 \bmod (5 \bmod 3)$
- iii. $13 \bmod (27 \bmod 4)$
- iv. $2^3 + 3 * (27 \bmod (25 \bmod 7))$
- v. $13/2 - 3 \bmod 2 - 3 \bmod 2$

5. Να παρουσιαστεί ο πίνακας τιμών και οι τιμές που θα εκτυπωθούν.

Αλγόριθμος Πίνακας_Τιμών

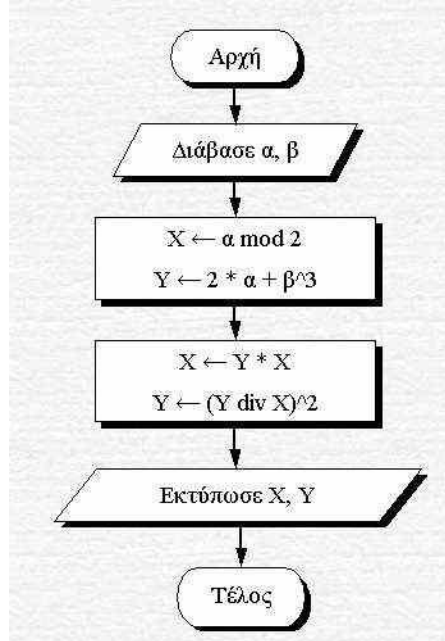
```

X ← 3
Y ← X ^ 3 - 4
Z ← Y div X
Εκτύπωσε Y, Z, X
X ← (X + Z) mod Y
Y ← (Y + Z) div X
Z ← X * Y - Z ^ 2

```

Εκτύπωσε Y, Z, X
Τέλος Πίνακας_Τιμών

6. Δίνεται το παρακάτω διάγραμμα ροής. Να γραφεί σε μορφή ψευδοκώδικα. Ποιός είναι ο πίνακας τιμών αν από το χρήστη εισαχθούν οι τιμές $\alpha=5$ και $\beta=-2$;



7. Να γραφεί αλγόριθμος ο οποίος θα διαβάζει τρεις αριθμούς και θα εκτυπώνει το άθροισμα, το γινόμενο και το μέσο όρο τους.
8. Στην αλυσίδα καταστημάτων **MakroX markets** οι τιμές των προϊόντων στις ετικέτες αναγράφονται χωρίς το ΦΠΑ. Να αναπτυχθεί ο αλγόριθμος που εκτελεί η ταμειακή μηχανή που θα διαβάζει την τιμή πληρωμής (χωρίς ΦΠΑ) και το ποσό που έδωσε ο πελάτης και θα υπολογίζει το πληρωτέο ποσό και τα ρέστα που δικαιούται ο πελάτης (συντελεστής ΦΠΑ 24%).
9. Να αναπτυχθεί αλγόριθμος που θα διαβάζει το όνομα ενός μαθητή της Γ' Λυκείου, τους βαθμούς του στα δυο τετράμηνα καθώς και τον γραπτό του βαθμό στις πανελλήνιες εξετάσεις και να υπολογίζει τον βαθμός πρόσβασης του μαθητή αυτού στο συγκεκριμένο μάθημα (Σημείωση: ο βαθμός πρόσβασης υπολογίζεται από την πράξη $70\% * \text{γραπτός βαθμός}$ και $30\% * \text{προφορικός βαθμός}$, όπου ο προφορικός βαθμός είναι ο μέσος όρος των βαθμών στα δυο τετράμηνα).
10. Να αναπτυχθεί αλγόριθμος ο οποίος θα διαβάζει έναν θετικό αριθμό x και θα υπολογίζει και θα εκτυπώνει την τιμή της ακόλουθης συνάρτησης.

$$f(x) = \frac{x - 4}{(x + 1)^3}$$

11. Η εταιρεία κινητής τηλεφωνίας ATeI αποφάσισε να δώσει δωρεά σε ιδρύματα ποσοστό 60% επί των εσόδων της από την αποστολή γραπτών μηνυμάτων μέσω του δικτύου της τις διακοπές των Χριστουγέννων, το κόστος κάθε μηνύματος είναι 0.072 €. Να αναπτύξετε αλγόριθμο που θα διαβάζει το πλήθος των μηνυμάτων που εστάλησαν τα Χριστούγεννα και να εκτυπώνει το ποσό που θα δοθούν στα ιδρύματα και τα κέρδη της εταιρείας.

12. Η ημερήσια αποζημίωση ενός εργαζόμενου είναι 35 €. Το ποσοστό των κρατήσεων για ασφάλεια είναι 11% και για φόρο 8.5%. Να αναπτυχθεί αλγόριθμος που θα διαβάζει το όνομα ενός εργαζομένου και τις ημέρες απασχόλησής του για τον τρέχοντα μήνα και θα εμφανίζει τις καθαρές αποδοχές για τον εργαζόμενο αυτό.
13. Να αναπτυχθεί αλγόριθμος που θα διαβάζει έναν τριψήφιο αριθμό και θα επιστρέφει το άθροισμα των ψηφίων του (για παράδειγμα για τον αριθμό 523 θα ισχύει $5 + 2 + 3 = 10$).

Β ΟΜΑΔΑ

1. Να διατυπώσετε σε λογικές εκφράσεις τις παρακάτω προτάσεις:
- Το α ανήκει στο διάστημα $[-5, 6)$
 - Το α είναι μικρότερο του 3 ή μεγαλύτερο του 15
 - Το α είναι ίσο με το β και το γ
 - Το α δεν έχει την τιμή 3
2. Ποιο είναι το λογικό αποτέλεσμα (αληθής ή ψευδής) από την εκτέλεση των παρακάτω πράξεων αν οι εξής μεταβλητές έχουν τιμές:
 $A = 10, B = 2, \Gamma = -4, \Delta = 9$ και $E = 1$
- $(A > B) \text{ Η } (\Delta = 10)$
 - $(\Delta \geq B) \text{ ΚΑΙ } (E < \Gamma)$
 - $\text{ΟΧΙ } (E \leq \Gamma) \text{ Η } (\Delta \leq \Gamma)$
 - $\text{ΟΧΙ } ((B \leq \Gamma) \text{ ΚΑΙ } (\Delta < 2))$

Δίνεται ο παρακάτω αλγόριθμος.

A) Ποιος είναι ο πίνακας τιμών;

B) Να σχηματίσετε το διάγραμμα ροής του παρακάτω αλγορίθμου:

Αλγόριθμος Πίνακας_Τιμών

```

X ← 2
Y ← X ^ 2 - 1
Z ← 2 * X + Y - 1
Αν (X > Y) τότε
    Y ← Z mod X
    Z ← X ^ 2

```

Αλλιώς

```

X ← Z mod Y
Z ← Y ^ 2

```

Τέλος_Αν

Εκτύπωσε X, Y, Z

Τέλος Πίνακας_Τιμών

3. A) Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τί θα εκτυπωθεί;
 B) Να σχηματίσετε το διάγραμμα ροής του παρακάτω αλγορίθμου:

Αλγόριθμος Πίνακας_Τιμών

```

α ← 3
β ← 1
γ ← 5

```

Αν $(\alpha \bmod 2 = 1)$ ή $(\beta \geq 2)$ τότε

$\gamma \leftarrow \gamma + 2$

Αν $(\gamma < \beta)$ τότε

$\alpha \leftarrow \alpha^3$

Αλλιώς

$\beta \leftarrow 4 * \beta$

Τέλος_Αν

Τέλος_Αν

$\alpha \leftarrow \alpha \bmod \beta$

$\beta \leftarrow \beta \bmod \gamma$

$\gamma \leftarrow \gamma \bmod \alpha$

Εκτύπωσε α, β, γ

Τέλος Πίνακας_Τιμών

4. Να σχηματίσετε το διάγραμμα ροής του παρακάτω αλγορίθμου:

Αλγόριθμος Διάγραμμα_Ροής

Διάβασε τιμή

Αν $(\text{τιμή} \leq 0)$ τότε

$\alpha_τ \leftarrow (-1) * \text{τιμή}$

Αλλιώς

$\alpha_τ \leftarrow \text{τιμή}$

Τέλος_Αν

Εκτύπωσε $\alpha_τ$

Τέλος Διάγραμμα_Ροής

5. Να σχηματίσετε το διάγραμμα ροής του παρακάτω αλγορίθμου:

Αλγόριθμος Διάγραμμα_Ροής

Διάβασε α

Αν $(\alpha \leq 2)$ τότε

$\text{τιμή} \leftarrow 15$

Αλλιώς_Αν $(\alpha \leq 10)$ τότε

$\text{τιμή} \leftarrow 11$

Αλλιώς_Αν $(\alpha \leq 20)$ τότε

$\text{τιμή} \leftarrow 9$

Αλλιώς

$\text{τιμή} \leftarrow 5$

Τέλος_Αν

Εκτύπωσε τιμή

Τέλος Διάγραμμα_Ροής

6. Να υλοποιήσετε τον παρακάτω αλγόριθμο με τη χρήση εμφωλευμένη δομής επιλογής.

Αλγόριθμος Μετατροπή_Δομές

Διάβασε α

Αν $(\alpha \leq 0)$ τότε

$\beta \leftarrow 0$

Αλλιώς_Αν $(\alpha \leq 5)$ τότε

$\beta \leftarrow 5$

Αλλιώς_Αν $(\alpha \leq 10)$ τότε

$\beta \leftarrow 10$

Αλλιώς

$\beta \leftarrow 100$

Τέλος_Αν

Εκτύπωσε β

Τέλος Μετατροπή_Δομές

7. Να αναπτυχθεί αλγόριθμος που θα διαβάζει έναν αριθμό που αντιστοιχεί στο βαθμό ενός μαθητή και θα εκτυπώνει μήνυμα αν είναι αποδεκτός (εντός των ορίων $[0, 20]$) ή όχι.

8. Να αναπτυχθεί αλγόριθμος ο οποίος θα διαβάζει έναν αριθμό x και θα υπολογίζει και θα εκτυπώνει την τιμή της ακόλουθης συνάρτησης:

$$f(x) = \frac{3x}{(x-1)^2}$$

9. Να αναπτυχθεί αλγόριθμος ο οποίος θα διαβάζει έναν αριθμό x και θα υπολογίζει και θα εκτυπώνει την τιμή της ακόλουθης συνάρτησης:

$$f(x) = \begin{cases} \frac{5}{(x-1)^2}, & x < 1 \\ 2, & x = 1 \\ \frac{5}{(x-1)^3}, & x > 1 \end{cases}$$

10. Η εταιρεία κινητής τηλεφωνίας Atel παρέχει στους συνδρομητές της λογότυπα και μελωδίες για τα κινητά τους. Για να καταβιβάσει κάποιος συνδρομητής στο κινητό του κάτι από τα παραπάνω πρέπει να εγγραφεί στο Atel club, με πάγιο μηνιαίο κόστος 0.90 €, και επιπλέον χρέωση 0.25 € για κάθε λογότυπα και 0.15 € για κάθε μελωδία που καταβιβάζετε στο κινητό του συνδρομητή. Να γραφεί αλγόριθμος που με δεδομένο το διαθέσιμο υπόλοιπο χρημάτων του κινητού ενός συνδρομητή, θα διαβάζει το σύνολο των λογότυπων και μελωδιών που κάποιος συνδρομητής ζήτησε για καταβίβαση στο κινητό του τον τελευταίο μήνα και αν το διαθέσιμο υπόλοιπο του το επιτρέπει να υπολογίζει το νέο υπόλοιπο.

11. Η εταιρεία κινητής τηλεφωνίας Atel παρέχει στους συνδρομητές της υπηρεσίες αποστολής μηνυμάτων σύμφωνα με τον παρακάτω πίνακα:

Είδος	Μήνυμα	Τιμή €
1.	Απλό sms	0.085
2.	Sms με ενσωματωμένο ήχο/εικόνα	0.67
3.	Εικονομήνυμα με επιπλέον στοιχεία	0.93
4.	Μήνυμα ενσωματωμένο βίντεο	1.25

Να γραφεί αλγόριθμος που να διαβάζει το διαθέσιμο υπόλοιπο χρημάτων του κινητού ενός συνδρομητή και θα διαβάζει το είδος του μηνύματος που επιθυμεί ο χρήστης να στείλει και - αν αυτό είναι εφικτό - θα επιστρέφει το νέο διαθέσιμο υπόλοιπο του συνδρομητή.

12. Το Internet Cafe της γειτονιάς σας έχει την εξής πολιτική χρέωσης: Τα πρώτα 30 λεπτά χρεώνονται 1.45 €, ενώ κάθε επόμενο 30λεπτο χρεώνεται προς 1.10 €. Πρέπει να επισημανθεί ότι μόλις περάσει έστω και ένα λεπτό χρεώνεται το 30λεπτο. Να αναπτυχθεί αλγόριθμος που θα διαβάζει το χρόνο (σε λεπτά) που παρέμεινε κάποιος πελάτης στο Internet Cafe και να εκτυπώνει το λογαριασμό.

13. Σύμφωνα με το Διατραπεζικό Σύστημα Συναλλαγών ΔΙΑΣ, κάποιος καταθέτης μπορεί να πραγματοποιήσει ανάληψη από κάποια άλλη τράπεζα πέραν αυτής που συνεργάζεται από ένα μηχάνημα ATM. Για την υπηρεσία αυτή υπάρχει χρέωση η οποία ισούται με το ένα εκατοστό του ποσού της ανάληψης. Η χρέωση αυτή δεν πρέπει να είναι μικρότερη από 1€ αλλά ούτε και να υπερβαίνει τα 3€. Να αναπτύξετε αλγόριθμος που με δεδομένο το διαθέσιμο υπόλοιπο του λογαριασμού του πελάτη, να διαβάσει το ποσό της ανάληψης από ένα ATM του ΔΙΑΣ, να ελέγχει αν μπορεί να πραγματοποιηθεί η συναλλαγή και να εκτυπώνει το υπόλοιπο του λογαριασμού και τη χρέωση που θα έχει ο πελάτης σύμφωνα με το ΔΙΑΣ.
14. Μια ιδιωτική εταιρεία αποφάσισε να εφαρμόσει ενιαία πολιτική στη μισθοδοσία του προσωπικού της. Έτσι, ο βασικός μισθός είναι 1200 €. Για τους αποφοίτους ΑΕΙ/ΤΕΙ υπάρχει επίδομα 20% ενώ αν κάποιος διαθέτει μεταπτυχιακό τίτλο τότε το επίδομα σπουδών γίνεται 29%. Επιπρόσθετα, κάθε υπάλληλος λαμβάνει χρονοεπίδομα 15% επί του βασικού μισθού αν βρίσκεται μέχρι και 5 χρόνια στην εταιρεία, 25% αν βρίσκεται μέχρι και 15 χρόνια ενώ 35% αν εργάζεται περισσότερα από 15 χρόνια στην εταιρεία. Να αναπτύξετε αλγόριθμο που θα διαβάσει το όνομα του μισθωτού, το επίπεδο σπουδών (1. βασική εκπαίδευση, 2. ΑΕΙ/ΤΕΙ και 3. Μεταπτυχιακές σπουδές) καθώς και τα έτη υπηρεσίας και στη συνέχεια να υπολογίζει και να εκτυπώνει τις μηνιαίες αποδοχές του.

Γ ΟΜΑΔΑ

1. α) Πόσες φορές θα εκτελεστούν οι επαναληπτικές δομές στα παρακάτω τμήματα αλγορίθμων;
β) Σχεδιάστε για κάθε ένα Διάγραμμα Ροής

A) $x \leftarrow 5$ Όσο ($x > 0$) επανάλαβε Εμφάνισε x $x \leftarrow x - 1$ Τέλος_Επανάληψης	B) $x \leftarrow 5$ Όσο ($x \geq 0$) επανάλαβε Εμφάνισε x $x \leftarrow x - 1$ Τέλος_Επανάληψης
Γ) $x \leftarrow -5$ Όσο ($x \geq 0$) επανάλαβε Εμφάνισε x $x \leftarrow x - 1$ Τέλος_Επανάληψης	Δ) $x \leftarrow 5$ Όσο ($x \geq 0$) επανάλαβε Εμφάνισε x $x \leftarrow x + 1$ Τέλος_Επανάληψης

2. Υπάρχει κάποιο λάθος στα παρακάτω τμήματα αλγορίθμων;

A) $S \leftarrow 0$ Για i από -3 μέχρι 3 Για j από 10 μέχρι 20 με_βήμα i $S \leftarrow S + 1$ Τέλος_Επανάληψης Τέλος_Επανάληψης Εκτύπωσε S	B) $S \leftarrow 0$ Για i από 2 μέχρι 5 Για j από 14 μέχρι i $S \leftarrow S + 2$ Τέλος_Επανάληψης Τέλος_Επανάληψης Εκτύπωσε S
---	--

3. α) Να σχηματίσετε τον πίνακα τιμών του παρακάτω τμήματος αλγορίθμου. Τι θα εκτυπωθεί τελικά;

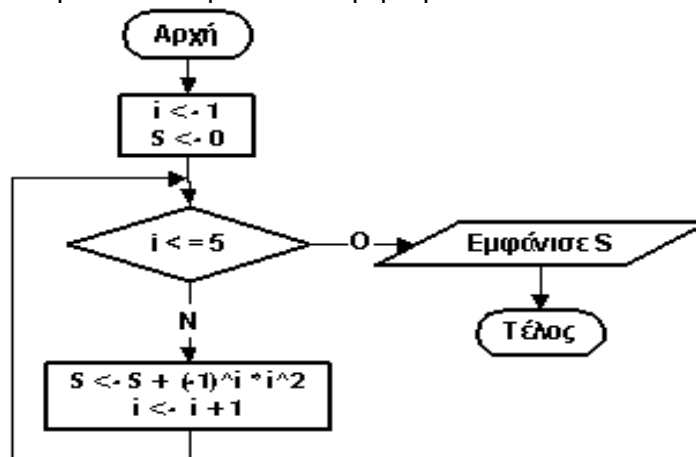
β) Σχεδιάστε για κάθε ένα Διάγραμμα Ροής

```
α ← 0
Όσο (α ≤ 22) επανάλαβε
    Για i από 1 μέχρι 3
        α ← α + i
    Τέλος_Επανάληψης
α ← α + 5
Τέλος_Επανάληψης
Εκτύπωσε α
```

4. Να μετατρέψετε το παρακάτω τμήμα αλγορίθμου χρησιμοποιώντας τις άλλες δυο δομές επανάληψης και να σχηματίσετε το διάγραμμα ροής.

```
α ← 2
β ← 3
Αρχή_Επανάληψης
    Εκτύπωσε β
    β ← β + 2
Μέχρις_Ότου (β > 11)
```

5. Να αναπαραστήσετε τον αλγόριθμο που αντιστοιχεί στο παρακάτω διάγραμμα ροής και να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί τελικά;



6. Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί τελικά;

Αλγόριθμος Πίνακας_Τιμών

```
α ← 2
β ← 1
Όσο (α ≥ β) και (α div 10 < 1) επανάλαβε
    α ← α ^ 2
    Αν (α div β > 2) τότε
        β ← β + 1
    Αλλιώς
        α ← α + 1
```

Τέλος_Αν
Τέλος_Επανάληψης
Εκτύπωσε α, β
Τέλος Πίνακας_Τιμών

7. α) Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί τελικά;
β) Σχεδιάστε για κάθε ένα Διάγραμμα Ροής

Αλγόριθμος Πίνακας_Τιμών

α ← 6

β ← 11

Αρχή_Επανάληψης

γ ← (α + β) div 2

Αν (γ > α) **τότε**

α ← γ - α

β ← β - γ

Αλλιώς

α ← 3 + α - γ

β ← γ - β

Τέλος_Αν

ποσότητα ← γ + α * β

Μέχρις_Ότου (ποσότητα < 0)

Εκτύπωσε α, β, γ

Τέλος Πίνακας_Τιμών

8. Να αναπτύξετε αλγόριθμο ο οποίος θα εκτυπώνει τις τιμές της συνάρτησης αν το χ παίρνει τιμές στο διάστημα [-0.5,5] με βήμα 0.05.

$$f(x) = \frac{x - 4}{(x + 1)^3}$$

9. Να αναπτύξετε αλγόριθμο ο οποίος θα υπολογίζει και θα εκτυπώνει το άθροισμα

$$1 + 2 + 3 + 4 + 5 + 6 + \dots + 1000.$$

10. Να αναπτυχθεί αλγόριθμος που θα διαβάζει έναν αριθμό N και να υπολογίζει τη σειρά

$$S = 5 + 3 - 9 + 27 - 81 + \dots \pm 3^N$$

11. Να αναπτυχθεί αλγόριθμος που θα εκτυπώνει τους τριψήφιους αριθμούς που είναι πολλαπλάσια του 7 καθώς και το πόσοι είναι οι αριθμοί αυτοί.

12. Να αναπτύξετε αλγόριθμο ο οποίος θα εντοπίζει και θα εκτυπώνει όλους τους τριψήφιους αριθμούς που το άθροισμα τετραγώνων των ψηφίων τους είναι μικρότερο από αυτούς (για παράδειγμα $131, 1^2 + 3^2 + 1^2 = 11 < 131$).

13. Να αναπτυχθεί αλγόριθμος που θα διαβάζει N αριθμούς και θα υπολογίζει και θα εκτυπώνει τον ελάχιστο.

14. Να αναπτυχθεί αλγόριθμος που θα διαβάζει έναν αριθμό (μεγαλύτερο του 0) και να υπολογίζει τη σειρά $S = 1 + 2^2 + 3^3 + 4^4 + \dots$ μέχρι να ξεπεράσει την τιμή του αριθμού αυτού και να εκτυπώνει το πλήθος των επαναλήψεων που χρειάστηκα.
15. Να αναπτύξετε αλγόριθμο που θα διαβάζει αριθμούς αγνώστου πλήθους και θα εκτυπώνει το μέσο όρο των θετικών. Η επαναληπτική διαδικασία να τερματίζεται όταν δοθεί ο αριθμός 0.
16. Να αναπτυχθεί αλγόριθμος που θα διαβάζει άγνωστο πλήθος αριθμών και θα εντοπίζει και εκτυπώνει το ποσοστό αυτών που είναι πολλαπλάσια του 5. Ο αλγόριθμος θα τερματίζεται όταν εισαχθεί ο αριθμός 0.
17. Από έρευνες έχει φανεί ότι μια κοινότητα μελισσών υπό κανονικές συνθήκες αναπτύσσεται με ρυθμό 3.8% ετησίως. Αν ένας μελισσοκόμος διαθέτει μελίσσια με συνολικό πληθυσμό 1200 μέλισσες σε πόσα έτη θα ξεπεράσει τη χωρητικότητα των κυψελών του που είναι 2000 μέλισσες;
18. Ο μισθός του κύριου Βασίλη είναι 1250 €, ενώ σύμφωνα με το μισθολόγιο αυξάνεται κατά 11% ετησίως. Κάθε μήνα έχει αποφασίσει να αποταμιεύει 9% του μισθού για το όνειρό του που είναι η αγορά φουσκωτού σκάφους.
19. Να αναπτύξετε αλγόριθμο που θα υπολογίζει και θα εκτυπώνει σε πόσους μήνες θα κατορθώσει να προβεί στην αγορά του φουσκωτού αξίας 7000 €.
20. Να αναπτυχθεί ο αλγόριθμος που εκτελείται στα διόδια. Για κάθε αυτοκίνητο που περνά να διαβάζεται ο τύπος του ("Φ" για φορτηγό, "Α" για αυτοκίνητο και "Μ" για μοτοσικλέτα) και να εκτυπώνεται το κόμιστρο. Ο αλγόριθμος να τερματίζεται όταν διαβάζει ως τύπο οχήματος "Τέλος" και να εκτυπώνει τις εισπράξεις της ημέρας. Πρέπει να επισημανθεί ότι το κόστος διέλευσης είναι 2.50 € για ένα φορτηγό, 1.40 για ένα αυτοκίνητο και 0.90 για μια μοτοσικλέτα.
21. Οι βαθμολογητές των γραπτών των πανελληνίων εξετάσεων βαθμολογούν με άριστα το 100, ενώ κάθε γραπτό διορθώνεται από 2 άτομα χωρίς να γνωρίζει ο ένας τη βαθμολογία του άλλου. Ωστόσο, αν μεταξύ των δυο βαθμολογιών παρατηρηθεί διαφορά μεγαλύτερη των 11 μορίων τότε το γραπτό διορθώνεται και από τρίτο βαθμολογητή και σε αυτήν την περίπτωση ο τελικός γραπτός βαθμός είναι ο μέσος όρος των 3 βαθμολογιών, διαφορετικά αν δεν υπάρξει αναβαθμολόγηση τελικός βαθμός θεωρείται ο μέσος όρος των 2 βαθμολογιών.

Δ ΟΜΑΔΑ

1. Να κρίνετε για την ορθότητά τους τις παρακάτω εντολές εκχώρησης τιμής

- | | |
|------------------------------------|---|
| i. τιμή $\leftarrow$ "τιμή" + 2 | iv. $u_i \leftarrow u_i + 3$ |
| ii. Διάβασε $\alpha + \beta$ | v. ποσό $\leftarrow \alpha * \beta + 5$ |
| iii. $\alpha + \beta \leftarrow 6$ | vi. Διάβασε $\leftarrow$ τιμή |

2. Δίνεται ο παρακάτω αλγόριθμος. Α) Να παρουσιαστεί ο πίνακας τιμών και οι τιμές που θα εκτυπωθούν. Β) Να δημιουργήσετε και το διάγραμμα ροής.

Αλγόριθμος Πίνακας_Τιμών1

$\kappa \leftarrow 3$

$\lambda \leftarrow \kappa + 2$

$\mu \leftarrow \lambda \wedge (\kappa - 1) - 3$

$\kappa \leftarrow \mu \text{ div } 3$

Εκτύπωσε κ, λ, μ

$\mu \leftarrow \lambda \wedge 3$

$\lambda \leftarrow \lambda + 2$

Εκτύπωσε μ, κ, λ

Τέλος Πίνακας_Τιμών3

3. Να αναπτυχθεί αλγόριθμος ο οποίος θα διαβάζει έναν θετικό αριθμό x και θα υπολογίζει και θα εκτυπώνει την τιμή της ακόλουθης συνάρτησης:

$$f(x) = \frac{x-7}{9+(3+x)^4} + (3-x)^3$$

4. Να αναπτυχθεί αλγόριθμος που θα διαβάζει τις κάθετες πλευρές ορθογωνίου τριγώνου και θα εκτυπώνει το εμβαδόν του και την υποτείνουσα του τριγώνου.
5. Κάθε εργαζόμενος της εταιρείας XYZ πληρώνεται με ημερομίσθιο 35 €, ενώ ο μισθός του υπόκειται σε κρατήσεις 12%. Να αναπτυχθεί αλγόριθμος που θα διαβάζει το ονοματεπώνυμο ενός εργαζομένου της εταιρείας XYZ, τις ημέρες που εργάστηκε τον περασμένο μήνα και να εκτυπώνει τις καθαρές αποδοχές του καθώς και το ποσό των κρατήσεων.
6. Σύμφωνα με την νέα φορολογική νομοθεσία η φορολόγηση των μικρομεσαίων εταιρειών είναι 25% επί των ετήσιων μικτών κερδών. Να αναπτύξετε αλγόριθμο που θα διαβάζει τα μικτά κέρδη μιας εταιρείας καθώς και τα έξοδα και να εκτυπώνει τα καθαρά κέρδη για την τρέχουσα χρονιά.
7. Τρεις φίλοι αποφάσισαν να καταθέσουν από κοινού ένα δελτίο στοιχήματος. Αποφάσισαν τα ενδεχόμενα κέρδη να μοιραστούν με βάση τη συμμετοχή τους στην πληρωμή του δελτίου. Να αναπτυχθεί αλγόριθμος που θα διαβάζει τα κέρδη που επέφερε το δελτίο καθώς και το ποσό που πλήρωσε κάθε ένας από τους τρεις και να εκτυπώνει το κέρδος που αναλογεί στον καθένα.

1. Να συμπληρώσετε τον παρακάτω πίνακα:

	A	Τελεστής	B	Αποτέλεσμα
i.	ΑΛΗΘΗΣ	ΚΑΙ	_____	ΑΛΗΘΗΣ
ii.	ΨΕΥΔΗΣ	Η	_____	ΑΛΗΘΗΣ
iii.	_____	ΚΑΙ	ΑΛΗΘΗΣ	ΨΕΥΔΗΣ
v.	ΨΕΥΔΗΣ	ΟΧΙ	_____	ΑΛΗΘΗΣ

2. Να συμπληρώσετε τον παρακάτω πίνακα με τις τιμές αληθής - ψευδής :

α	β	γ	$(\alpha \bmod 2 = 0)$ ή $(\beta \leq 3)$	$(\alpha \leq \beta)$ και $(\beta \geq \gamma)$	όχι $(\alpha > \beta)$ ή $(\gamma \geq 2)$	$(\gamma \geq \alpha)$ και $(\beta \div 2 = 1)$
2	-1	0	_____	_____	_____	_____
-3	1	-4	_____	_____	_____	_____

3. Δίνεται ο παρακάτω αλγόριθμος. **α)** Να παρουσιαστεί ο πίνακας τιμών και οι τιμές που θα εκτυπωθούν. **β)** Σχεδιάστε Διάγραμμα Ροής.

Αλγόριθμος Πίνακας_Τιμών

$\alpha \leftarrow 3$

$\beta \leftarrow \alpha^2$

$\gamma \leftarrow 5$

Αν $(\beta \bmod 2 = 1)$ **και** $(\beta \geq 5)$ **τότε**

$\gamma \leftarrow \gamma + 2$

Αν $(\gamma > \alpha)$ **τότε**

$\alpha \leftarrow \alpha^3$

$\gamma \leftarrow \gamma^2$

Αλλιώς

$\alpha \leftarrow \alpha / 4$

$\gamma \leftarrow \gamma / 2$

Τέλος_Αν

Εκτύπωσε α, β, γ

Τέλος_Αν

$\alpha \leftarrow \alpha \div \beta$

$\beta \leftarrow \beta \bmod \gamma$

$\gamma \leftarrow \gamma \bmod \alpha$

Εκτύπωσε α, β, γ

Τέλος Πίνακας_Τιμών

4. Να αναπτυχθεί αλγόριθμος ο οποίος θα διαβάζει έναν αριθμό x και θα υπολογίζει και θα εκτυπώνει την τιμή της ακόλουθης συνάρτησης.

$$f(x) = \begin{cases} 15x^3, & x < 0 \\ 12, & x = 0 \\ \frac{5x}{(x+3)^2}, & x > 0 \end{cases}$$

5. Να αναπτυχθεί αλγόριθμος που θα διαβάσει τις δικαιολογημένες και τις αδικαιολόγητες απουσίες ενός μαθητή και να αποφαινεται αν ο μαθητής απορρίπτεται λόγω απουσιών ή παραπέμπει σε έλεγχο της βαθμολογίας (σημειώνεται ότι τα όρια των απουσιών είναι 64 δικαιολογημένες και 50 αδικαιολόγητες).
6. Να αναπτυχθεί αλγόριθμος που θα διαβάσει 3 αριθμούς και θα τους εκτυπώνει στη σειρά από τον μικρότερο στον μεγαλύτερο.
7. Η εταιρεία καρτοκινητής τηλεφωνίας Tel προσφέρει σύνδεση στο τηλεφωνικό δίκτυό της χωρίς πάγιο, συμβόλαιο και άλλες δεσμεύσεις. Η χρέωση κάθε κλήσης πραγματοποιείται κάθε 30 δευτερόλεπτα (μια κλήση που διαρκεί 33" θα στοιχίσει όσο μια των 60"). Το κόστος για κάθε μισό λεπτό είναι 0.09 €, ωστόσο αν μια κλήση διαρκέσει πέραν των 3 λεπτών η χρέωση του επιπλέον χρόνου είναι 0.065 €/30" κλήσης. Να αναπτυχθεί αλγόριθμος που θα διαβάσει το όνομα του πελάτη και τη χρονική διάρκεια μιας κλήσης και την χρέωση της κλήσης αυτής. Σημειώνεται ότι στα παραπάνω ποσά υπάρχει επιβάρυνση ΦΠΑ 24%.
8. Τα ταχυδρομικά τέλη για την ταχυδρομική εταιρεία Courier υπολογίζονται ως εξής:
 - για φακέλους μικρότερους των 20 γραμμαρίων, 1.20 €
 - για φακέλους μικρότερους των 150 γραμμαρίων, 2 €
 - για φακέλους από 150 γραμμάρια και άνω 0.015 € για κάθε γραμμάριο.

Να αναπτυχθεί αλγόριθμος που θα διαβάσει το βάρος του φακέλου που θα αποσταλεί και θα υπολογίζει και θα εκτυπώνει την χρέωση.

9. Σύμφωνα με τον τελευταίο νόμο του Υπουργείου Οικονομικών οι μεγαλοοφειλέτες μπορούν να προσέλθουν στις κατά τόπους εφορίες και να προβούν σε διακανονισμό για την πληρωμή των οφειλών τους. Έτσι, αν κάποιος χρωστά περισσότερα από 500.000 € θα έχει έκπτωση 20%, ενώ διαφορετικά θα έχει έκπτωση 30%. Αν κάποιος πληρώσει μετρητοίς τότε θα έχει επιπλέον έκπτωση 5% επί του ποσού της οφειλής. Να αναπτυχθεί αλγόριθμος που θα διαβάσει το όνομα του οφειλέτη, το ποσό της οφειλής του καθώς και το πλήθος των δόσεων που θα εξοφλήσει (δόσεις = 1 σημαίνει πληρωμή μετρητοίς) και θα εκτυπώνει το ποσό κάθε δόσης στην εφορία.
10. Η βιοτεχνία κατασκευής ενδυμάτων CLOTHES ακολουθεί την εξής τιμολογιακή πολιτική για την πώληση (χονδρικής) των προϊόντων της: Για παραγγελία άνω των 800 τεμαχίων, 17 €/τεμάχιο, για παραγγελία άνω των 350 τεμαχίων, 21 €/τεμάχιο και για παραγγελία άνω των 150 τεμαχίων, 25 €/τεμάχιο, παραγγελίες μικρότερες των 150 τεμαχίων δεν γίνονται δεκτές. Να αναπτυχθεί αλγόριθμος που θα διαβάσει το όνομα του πελάτη και των πλήθος των τεμαχίων της παραγγελίας και θα εκτυπώνει το κόστος της παραγγελίας. Σημειώνεται πως αν το κόστος της παραγγελίας υπερβεί τα 1500 € πραγματοποιείται έκπτωση 9%.
11. Στο εργοστάσιο AMANAE συνέβη εργατικό ατύχημα με αποτέλεσμα τον σοβαρό τραυματισμό ενός εργάτη. Οι συνάδελφοί του αποφάσισαν να πραγματοποιήσουν έρανο για τα έξοδα νοσηλείας του. Αποφάσισαν ο καθ' ένας να προσφέρει ανάλογα με τις αποδοχές του, ως εξής:

Μηνιαίες απολαβές (σε €)	Συντελεστής %
Μέχρι 800	4
801 - 1200	8
1201 και άνω	12

Να αναπτύξετε αλγόριθμο που θα διαβάσει το όνομα και το μισθό ενός εργαζομένου του εργοστασίου, να υπολογίζει και να εκτυπώνει το ποσό που θα χορηγήσει ως δωρεά καθώς και τον μισθό που τελικά θα εισπράξει τον τρέχοντα μήνα.

12. Σύμφωνα με το νέο φορολογικό νόμο ο συντελεστής φόρου για τους ιδιώτες φορολογούμενους απεικονίζεται στον παρακάτω πίνακα:

Εισόδημα (σε €)	Συντελεστής %
Μέχρι 15.000	0
15.001 – 40.000	8
40.001 – 65.000	11
65.001 - 80.000	14
80.001 και άνω	18

Ωστόσο, αν ο φορολογούμενος υποβάλλει την φορολογική του δήλωση από το διαδίκτυο (σύστημα **TAXIS**), έχει έκπτωση 10%. Να αναπτύξετε αλγόριθμο που θα διαβάζει το όνομα ενός φορολογούμενου που υπέβαλε τη δήλωσή του από το διαδίκτυο, το εισόδημα που δήλωσε και θα υπολογίζει και θα εκτυπώνει το ποσό που πρέπει να εισπραχθεί από την εφορία καθώς και την έκπτωση.

(Να σημειωθεί ότι η ο φόρος υπολογίζεται κλιμακωτά).

Ζ ΟΜΑΔΑ

1. Να σχηματίσετε τον πίνακα τιμών για τα παρακάτω τμήματα αλγορίθμων. Τι θα εκτυπωθεί τελικά;

A)	B)	Γ)
$\alpha \leftarrow 3$	$\alpha \leftarrow 3$	$\alpha \leftarrow 3$
$\beta \leftarrow -5$	$\beta \leftarrow -5$	Για β από -5 μέχρι 0 με_βήμα 2
Όσο $\beta \leq 0$ επανάλαβε	Όσο $\beta \leq 0$ επανάλαβε	$\alpha \leftarrow \alpha + \beta - 1$
$\beta \leftarrow \beta + 2$	$\alpha \leftarrow \alpha + \beta - 1$	Τέλος_Επανάληψης
$\alpha \leftarrow \alpha + \beta - 1$	$\beta \leftarrow \beta + 2$	Εκτύπωσε α
Τέλος_Επανάληψης	Τέλος_Επανάληψης	
Εκτύπωσε α	Εκτύπωσε α	

2. Πόσες επαναλήψεις θα εκτελέσει ο κάθε ένας από τους παρακάτω αλγορίθμους και τι θα εκτυπωθεί τελικά;

A)	B)
Για i από 1 μέχρι 4	Για i από 1 μέχρι 7 με_βήμα 3
Εκτύπωσε "*"	Εκτύπωσε "*"
Τέλος_Επανάληψης	Τέλος_Επανάληψης
Για i από 6 μέχρι 4 με_βήμα -0.5	Για i από 2 μέχρι 6 με_βήμα -2
Εκτύπωσε "*"	Εκτύπωσε "*"
Τέλος_Επανάληψης	Τέλος_Επανάληψης

3. α) Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί τελικά;

β) Σχεδιάστε Διάγραμμα Ροής.

Αλγόριθμος Πίνακας_Τιμών

$\alpha \leftarrow 3$

$\beta \leftarrow 0$

Για i από 51 μέχρι 10 με_βήμα -11

$\alpha \leftarrow \alpha + 2$

Αν $(\alpha > 4)$ τότε
 $\beta \leftarrow \beta + i \text{div } \alpha$
Αλλιώς
 $\beta \leftarrow \beta - i$
Τέλος_Αν
Τέλος_Επανάληψης
 $\alpha \leftarrow \alpha - \beta$
Εκτύπωσε α, β
Τέλος Πίνακας_Τιμών

4. α) Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τί θα εκτυπωθεί τελικά;
 β) Σχεδιάστε Διάγραμμα Ροής.

Αλγόριθμος Πίνακας_Τιμών
 $\beta \leftarrow 0$
 $\gamma \leftarrow 2$
Για i από 1 μέχρι 3
 $\alpha \leftarrow 20 * i$

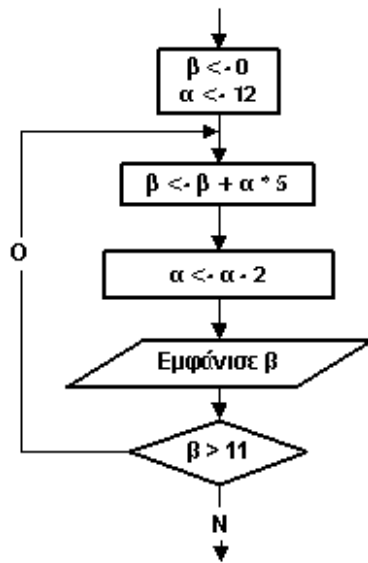
Αρχή_Επανάληψης
 $\beta \leftarrow \beta + \alpha \text{ div } 4$
 $\alpha \leftarrow \gamma + \alpha$
Μέχρις_Ότου $(\beta > 20 * i)$
 $\beta \leftarrow (3 * \alpha) \text{ div } 2$
 $\gamma \leftarrow \alpha \text{ div } \gamma$
Τέλος_Επανάληψης
Εμφάνισε α, β
Τέλος Πίνακας_Τιμών1

5. α) Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τί θα εκτυπωθεί τελικά;
 β) Σχεδιάστε Διάγραμμα Ροής.

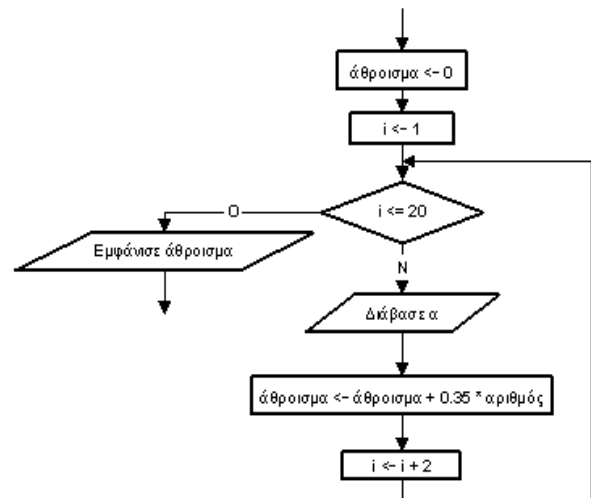
Αλγόριθμος Πίνακας_Τιμών
 $\alpha \leftarrow 321$
 $\beta \leftarrow 7$
Όσο $\alpha \text{ div } 10 > 0$ επανάλαβε
 $\alpha \leftarrow \alpha \text{ div } \beta$
Αν $(\beta > \alpha \text{ div } 2)$ τότε
 $\beta \leftarrow \beta * 3$
Αλλιώς
 $\beta \leftarrow \beta + 3$
Τέλος_Αν
Εκτύπωσε α, β
Τέλος_Επανάληψης
Τέλος Πίνακας_Τιμών

6. Να μεταφέρετε τα παρακάτω διαγράμματα ροής σε μορφή ψευδοκώδικα

A)



B)



7. Να αναπτύξετε αλγόριθμο ο οποίος θα εκτυπώνει τις τιμές της ακόλουθης συνάρτησης αν το x παίρνει ακέραιες τιμές στο διάστημα $[-50, 50]$

$$f(x) = \frac{1}{x^2 - 3x + 2}$$

8. Να αναπτυχθεί αλγόριθμος που θα εκτυπώνει όλες τις ακέραιες λύσεις της εξίσωσης $2x^2 + 3y - 7 = 0$ όταν το x παίρνει ακέραιες τιμές στο διάστημα $[-20, 20]$.

9. Να αναπτυχθεί αλγόριθμος που θα διαβάζει έναν αριθμό (μεγαλύτερο του 0) και να υπολογίζει τη σειρά:

$$S = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \pm \frac{1}{N}$$

10. Να αναπτύξετε αλγόριθμο οποίος θα υπολογίζει τη σειρά:

$$\alpha. S = 5^2 + 10^2 + 15^2 + 20^2 + \dots + 95^2 + 100^2$$

$$\beta. S = 5^2 + 10^2 + 15^2 + 20^2 + \dots$$

μέχρι το άθροισμα να ξεπεράσει το 100^4

11. Να αναπτυχθεί αλγόριθμος που θα εκτυπώνει το άθροισμα των τετραγώνων των διψήφιων άρτιων αριθμών

12. Να αναπτυχθεί αλγόριθμος που θα εμφανίζει τους διψήφιους άρτιους ακραίους.

13. Να αναπτυχθεί αλγόριθμος που θα εμφανίζει τους θετικούς τριψήφιους ακραίους που είναι πολλαπλάσια του 7, καθώς και το πλήθος των αριθμών αυτών.

14. Να αναπτύξετε αλγόριθμο που θα βρίσκει και θα εμφανίζει όλους τους θετικούς τριψήφιους ακραίους, των οποίων η τιμή είναι ίση με το άθροισμα των κύβων των ψηφίων τους (αριθμοί Armstrong).

15. Να δοθεί αλγόριθμος που θα υπολογίζει και τυπώνει το πλήθος και το άθροισμα όλων των άρτιων αριθμών από το 1 ως το 100.

16. Να γραφεί αλγόριθμος ο οποίος θα διαβάζει το φύλο , 'Α' ή 'Γ', 100 ανθρώπων και θα εμφανίζει το πλήθος των ανδρών και το πλήθος των γυναικών.
17. Σε ένα αγώνα ρίψης ακοντίου , διεξάγεται ο προκριματικός γύρος με τη συμμετοχή 14 αθλητών. Στην τελική φάση προκρίνονται όσοι αθλητές επιτύχουν επίδοση άνω των 80 μέτρων. Να γραφεί αλγόριθμος ο οποίος :
 - α) να διαβάζει το όνομα και την επίδοση κάθε αθλητή, να υπολογίζει και να εμφανίζει τα ονόματα και το πλήθος των αθλητών που πέρασαν το όριο.
 - β) να εμφανίζει το όνομα του αθλητή που πλησίασε πιο κοντά από όλους τα 70 μέτρα.
18. Να αναπτυχθεί αλγόριθμος που θα διαβάζει 500 θετικούς ακεραίους και θα εκτυπώνει το ποσοστό των αριθμών που είναι διψήφιοι.
19. Να αναπτυχθεί αλγόριθμος που θα διαβάζει χαρακτήρες κειμένου, μέχρι να συναντήσει τους χαρακτήρες «.» ή «!». Ο αλγόριθμος θα εμφανίζει το πλήθος των χαρακτήρων που διάβασε καθώς και το πλήθος εμφανίσεων του χαρακτήρα «α» (οι χαρακτήρες «.» και «!» δεν καταμετρούνται).
20. Να αναπτύξετε αλγόριθμο που θα διαβάζει άγνωστο πλήθος αριθμών μέχρι το άθροισμά τους να ξεπερνά την τιμή 500. Θα εκτυπώνεται το πλήθος των αριθμών που διαβάστηκαν.
21. Να αναπτύξετε αλγόριθμο που θα εκτυπώνει όλους τους τριψήφιους θετικούς ακέραιους που έχουν όλα τους τα ψηφία περιττά (για παράδειγμα οι αριθμοί 157, 399, 911).
22. Να αναπτύξετε αλγόριθμο που θα εκτυπώνει τους τριψήφιους θετικούς ακέραιους των οποίων τουλάχιστον ένα ψηφίο έχει την τιμή 3.
23. Να αναπτυχθεί αλγόριθμος που θα διαβάζει άγνωστο πλήθος αριθμών μέχρι να διαβάσει 100 αριθμούς ή μέχρι να εισαχθούν 20 αρνητικοί αριθμοί και να εκτυπώνεται το ποσοστό θετικών και αρνητικών αριθμών που εισήχθησαν.
24. Να αναπτυχθεί αλγόριθμος που θα διαβάζει άγνωστο πλήθος αριθμών μέχρι το άθροισμά τους να ξεπερνά την τιμή 500 να εκτυπώνεται το πλήθος των αριθμών που διαβάστηκαν.
25. Ο συντελεστής ΦΠΑ για τα προϊόντα του εμπορικού καταστήματος "Νέα ενδύματα" σε συσχέτιση με τον κωδικό τους δίνεται στον επόμενο πίνακα :

Κωδικός προϊόντος	Συντελεστής ΦΠΑ %
Z11	8
Γ78	12
O11	18

Να αναπτυχθεί αλγόριθμος που θα διαβάζει για μια παραγγελία τους κωδικούς των προϊόντων, τα τεμάχια και την τιμή τους (χωρίς ΦΠΑ) και να εκτυπώνει το πληρωτέο ποσό. Πρέπει να επισημανθεί ότι η επανάληψη θα τερματίζεται όταν δοθεί ο κωδικός "ΤΕΛΟΣ".

26. Σύμφωνα με το νέο φορολογικό νόμο ο συντελεστής φόρου για τους ιδιώτες φορολογούμενους απεικονίζεται στον παρακάτω πίνακα (κλιμακωτός υπολογισμός):

Εισόδημα (σε €)	Συντελεστής %
Μέχρι 12.0000	
12.001 - 30.000	8
30.001 - 50.000	12
50.001 - 80.000	16
80.001 και άνω	25

Ταυτόχρονα, υπάρχουν φοροελαφρύνσεις ανάλογα με τον αριθμό των παιδιών του φορολογούμενου, σύμφωνα με τον ακόλουθο πίνακα:

αριθμός παιδιών	Ποσό έκπτωσης
1 - 3	500 ανά παιδί
4 και άνω	3000

Να γραφεί αλγόριθμος που για κάθε έναν από τους 10.000.000 Έλληνες φορολογούμενους θα διαβάζει το όνομα, το εισόδημα που δήλωσε στην εφορία και το πλήθος παιδιών του και θα υπολογίζει και θα εκτυπώνει το φόρο που πρέπει να πληρωθεί. Ο αλγόριθμος θα επιστρέφει και το συνολικό ποσό που θα εισπράξει η εφορία τη φετινή χρονιά. (ο φόρος υπολογίζεται **κλιμακωτά**).

27. Οι υπάλληλοι μίας εταιρείας συμφώνησαν για το μήνα Δεκέμβριο να κρατηθούν από το μισθό τους δύο ποσά, ένα για την ενίσχυση του παιδικού χωριού SOS και ένα για την ενίσχυση των σκοπών της UNICEF. Ο υπολογισμός του ποσού των εισφορών εξαρτάται από τον αρχικό μισθό του κάθε υπαλλήλου και υπολογίζεται με βάση τα παρακάτω όρια μισθών:

Μισθός (€)	Εισφορά % SOS	Εισφορά % UNICEF
Έως 500	5	4
501 - 800	7.5	6
801 - 1100	9.5	8
Μεγαλύτερο από 1100	12	11

Να γραφεί αλγόριθμος που να διαβάζει για τους 500 υπαλλήλους της εταιρείας το όνομα και τον μισθό τους και να εκτυπώνει το καθαρό ποσό που θα πάρει ο κάθε ένας ως μισθό τον μήνα Δεκέμβριο. Ο αλγόριθμος πρέπει επίσης τελικά να εκτυπώνει το συνολικό ποσό που θα δοθεί στο χωριό SOS καθώς και το ποσό που θα δοθεί στη UNICEF.

28. Η χρέωση (**κλιμακωτή**) στους λογαριασμούς της **HellasTel** είναι η εξής:

Πάγιο:	35 €	
Αστικές μονάδες:	0.025 € ανά μονάδα	
Υπεραστικές μονάδες:	0 - 150	0.072 € ανά μονάδα
	151 - 500	0.025 € ανά μονάδα
	501 -	0.14€ ανά μονάδα

Να αναπτυχθεί αλγόριθμος που θα διαβάζει επαναληπτικά το όνομα ενός συνδρομητή, τις αστικές και τις υπεραστικές μονάδες που κατανάλωσε και να εκτυπώνει το ποσό του λογαριασμού του, μέχρι να διαβαστεί ως όνομα του **"ΤΕΛΟΣ"**. Ο αλγόριθμος πρέπει να επιστρέφει στο τέλος το συνολικό ποσό εισπράξεων της εταιρίας.

29. Να αναπτύξετε αλγόριθμο που να υπολογίζει και να εμφανίζει την παράσταση:

$$= \frac{(-101) * (-99) * ... * (99) * (101)}{(-102) * (-100) * ... * (100) * (102)}$$

30. Σε ένα κατάστημα αποφασίστηκε λόγω των Χριστουγέννων να προσφέρεται έκπτωση στους πελάτες ανάλογα με το ποσό των αγορών και με βάση τον παρακάτω πίνακα:

Αγορές (€)	Έκπτωση
Μέχρι 100.00	5%
100.01 – 300.00	10%
Άνω των 300.00	15%

Να γραφεί αλγόριθμος που :

α) να διαβάζει το ποσό αγοράς ενός πελάτη.

β) να υπολογίζει το ποσό της έκπτωσης που του αναλογεί.

γ) να υπολογίζει και να εμφανίζει το ποσό που θα πληρωθεί στο κατάστημα καθώς και την έκπτωση που αναλογεί.

(Σημείωση : να θεωρηθεί ότι το ποσό αγοράς είναι θετικός αριθμός).

31. Σε ένα εργοστάσιο οι εργαζόμενοι δουλεύουν 35 ώρες ανά εβδομάδα. Αν κάποιος εργαζόμενος δουλέψει υπερωρίες (υπερβεί τις 35 ώρες) πληρώνεται επιπλέον όπως φαίνεται στον παρακάτω πίνακα:

Ώρες Εργασίας	Ωρομίσθιο
0 ως 35 ώρες	1,50 €/ώρα
36 ως 45 ώρες	2,00 €/ώρα
46 και άνω	3,50 €/ώρα

Να γραφεί αλγόριθμος που διαβάζει τις ώρες εργασίας ενός εργαζόμενου στη διάρκεια μιας εβδομάδας, και υπολογίζει και εμφανίζει τις αποδοχές του.

(Σημείωση : να θεωρηθεί ότι οι ώρες εργασίες είναι πάντα θετικός αριθμός).

32. Το υπουργείο οικονομικών επιβάλλει τέλη κυκλοφορίας στα ιδιωτικής χρήσεως επιβατηγά οχήματα (αυτοκίνητα και δίκυκλα) ανάλογα με τον κυβισμό τους σύμφωνα με τον παρακάτω πίνακα :

Τύπος Οχήματος	Κυβισμός	Τέλος Κυκλοφορίας
Δίκυκλα	Μέχρι 500 cc	50 €
Δίκυκλα	Από 501cc και άνω	90 €
Αυτοκίνητα	Μέχρι 1400 cc	250 €
Αυτοκίνητα	Από 1401 μέχρι 1800 cc	360 €
Αυτοκίνητα	Από 1801 μέχρι 2000 cc	500 €
Αυτοκίνητα	Από 2000cc και άνω	1000 €

Να γραφεί αλγόριθμος ο οποίος :

α) να διαβάζει τον τύπο του οχήματος (Δ = Δίκυκλο, Α = Αυτοκίνητο) και τον κυβισμό του.

β) να υπολογίζει τα αντίστοιχα τέλη κυκλοφορίας του οχήματος.

γ) να εμφανίζει το ποσό που αναλογεί.

33. Ένα super market κάνει προσφορές στους κατόχους κάρτας μέλους ανάλογα με τους πόντους που συγκεντρώνουν στις αγορές τους. Για αγορά αξίας 3€ αντιστοιχεί ένας πόντος και μόλις συμπληρωθούν 200 πόντοι τότε γίνεται έκπτωση 6€ στο ποσό αγορών και μηδενίζεται ο αριθμός των πόντων στην κάρτα. Να γραφεί αλγόριθμος ο οποίος :

- α) διαβάσει το ποσό αγορών σε ευρώ για έναν πελάτη.
- β) διαβάσει την πληροφορία αν είναι κάτοχος κάρτας μέλους ή όχι.
- γ) σε περίπτωση που ο πελάτης είναι κάτοχος κάρτας μέλους τότε.
 - 1) να διαβάσει τους πόντους που είναι καταγεγραμμένοι στην κάρτα πριν την συγκεκριμένη αγορά.
 - 2) να υπολογίζει τους πόντους που αναλογούν από τη συγκεκριμένη αγορά.
 - 3) να υπολογίζει το σύνολο των πόντων στη συγκεκριμένη αγορά .
 - 4) να υπολογίζει και να εμφανίζει το ποσό που θα πληρώσει καθώς και το σύνολο των πόντων που υπάρχουν στην κάρτα μετά την αγορά.
- δ) Σε περίπτωση που ο πελάτης δεν έχει κάρτα να εμφανίζει το ποσό των αγορών που θα πληρώσει.

Σημείωση : Να θεωρήσετε ότι οι τιμές της μεταβλητής για την κάρτα μέλους είναι αλφαριθμητική και παίρνει αποκλειστικά τις τιμές «**Ναι**» ή «**Όχι**».

34. Μία τράπεζα εφαρμόζει το πρόγραμμα «ΕΠΙΣΤΡΟΦΗ» για τους κατόχους των πιστωτικών καρτών της. Το πρόγραμμα αυτό συγκεντρώνει Ευρώ από το ποσό αγορών που γίνεται σε συνεργαζόμενα καταστήματα ένα συγκεκριμένο ποσοστό % (**BONUS**) επί του ποσού αγοράς το οποίο διαφέρει ανά κατάστημα . Αν το ποσό επιστροφής που έχει συγκεντρωθεί είναι μη μηδενικό μπορεί να αφαιρεθεί από το ποσό αγοράς που γίνεται εκείνη τη στιγμή αν το θελήσει ο πελάτης, διαφορετικά αθροίζεται στο προηγούμενο ποσό. Να γραφεί αλγόριθμος ο οποίος :

- α) διαβάσει την πληροφορία (**αληθής ή ψευδής**) αν το κατάστημα που γίνεται μια αγορά είναι συνεργαζόμενο με την τράπεζα
- β) σε περίπτωση που το κατάστημα είναι συνεργαζόμενο τότε
 - 1) διαβάσει το ποσό αγοράς.
 - 2) διαβάσει το ποσοστό με το οποίο συμμετέχει το κατάστημα στο πρόγραμμα επιστροφής.
 - 3) διαβάσει το ποσό που έχει συγκεντρώσει πριν τη συγκεκριμένη αγορά στην κάρτα ο κάτοχος.
 - 4) διαβάσει μια λογική μεταβλητή «**ΕΞΑΡΓΥΡΩΣΗ**» και αποθηκεύει σε αυτή την αντίστοιχη επιθυμία του πελάτη (αληθής ή ψευδής).
 - 5) υπολογίζει και εκτυπώνει:
 - i) το ποσό που θα πληρωθεί στο κατάστημα.
 - ii) το ποσό (**BONUS**) που απομένει στην πιστωτική κάρτα.
- γ) σε περίπτωση που το κατάστημα δεν συνεργάζεται με την τράπεζα τότε να εκτυπώνεται το ποσό αγοράς.

EXTRA ΆΛΥΤΕΣ ΑΣΚΗΣΕΙΣ

- 1 Ο ταμίας ενός καταστήματος καταχωρεί στην ταμειακή μηχανή τον κωδικό, την τιμή και την ποσότητα του προϊόντος που θέλει να αγοράσει ο πελάτης. Όταν ολοκληρωθεί η διαδικασία αγοράς, ο ταμίας θα πρέπει να εισάγει τον κωδικό 0. Να γράψετε πρόγραμμα που θα υλοποιεί την παραπάνω διαδικασία και τελικά θα εμφανίζει το συνολικό ποσό της αγοράς και το πλήθος των προϊόντων που αγοράστηκαν.
- 2 Να γράψετε πρόγραμμα το οποίο θα διαβάζει αριθμούς και θα εμφανίζει το μέσο όρο των περιττών απ' αυτούς τους αριθμούς. Η παραπάνω διαδικασία να ολοκληρώνεται όταν θα έχουν διαβαστεί 100 περιττοί αριθμοί.
- 3 Να γίνει αλγόριθμος που θα διαβάζει τα ονόματα δύο παικτών του μπάσκετ και το ύψος τους σε εκατοστά. Στην συνέχεια να εμφανίζει το όνομα του ψηλότερου σε μήνυμα της μορφής: "Ο ψηλότερος παίκτης είναι ο _____".
- 4 Ένας 6ψήφιος κωδικός θεωρείται έγκυρος αν ισχύουν τα ακόλουθα:
 - 1) Το άθροισμα του 1ου και του 2ου ψηφίου είναι ίσο με το 3ο ψηφίο.
 - 2) το υπόλοιπο της διαίρεσης του 3ου με το 4ο ψηφίο είναι ίσο με το 5ο ψηφίο μείον 2.
 - 3) και η διαφορά του 6ου με το 2ο ψηφίο είναι ίσο με 3.
 Να γίνει αλγόριθμος που θα διαβάζει έναν εξαψήφιο αριθμό και θα ελέγχει αν ο κωδικός είναι έγκυρος ή όχι.
- 5 Να αναπτύξετε αλγόριθμο που θα διαβάζει για μια ομάδα 2.000 ανθρώπων το όνομα, το φύλο, το ύψος και το βάρος και θα εκτυπώνει:
 1. το όνομα του πιο βαρύ άντρα,
 2. το όνομα της πιο ψηλής γυναίκας, και
 3. το μέσο ύψος των γυναικών.

Παρατήρηση: να θεωρήσετε ότι στο δείγμα υπάρχει τουλάχιστον ένας άνδρας και μια γυναίκα.
- 6 Να γίνει αλγόριθμος που θα διαβάζει την ένδειξη ενός θερμομέτρου (σε βαθμούς Κελσίου) και θα εμφανίζει τα εξής μηνύματα:

"Φυσιολογικός" αν η θερμοκρασία είναι από 35,5 μέχρι 37.

"Ζεστός" αν η θερμοκρασία είναι πάνω από 37 μέχρι 38.

"Άρρωστος" αν η θερμοκρασία είναι πάνω από 38 .
- 7 Ένας έμπορος ελαστικών διαθέτει τα ελαστικά του σε χονδρική πώληση, σύμφωνα με τον επόμενη πολιτική:

Αριθμός ελαστικών Χρέωση

1 - 100	58€ / τεμάχιο
101 - 200	53€ / τεμάχιο
201 - 300	51€ / τεμάχιο
> 300	49€ / τεμάχιο

Επιπρόσθετα ο έμπορος χρεώνει την μεταφορά των ελαστικών στο συνεργαζόμενο κατάστημα σύμφωνα με την επόμενη πολιτική:

Βάρος	Χρέωση
< 1 τόνος	0,2€ / κιλό
1 – 3 τόνους	0,15€ / κιλό
> 3 τόνους	0,10€ / κιλό

Η χρέωση των μεταφορικών γίνεται κλιμακωτά. Δεδομένου ότι κάθε ελαστικό ζυγίζει περίπου 3,5 κιλά, να γίνει αλγόριθμος που θα διαβάσει τον αριθμό ελαστικών που θα παραγγείλει κάποιο κατάστημα και θα εκτυπώνει, το κόστος της παραγγελίας, το κόστος των μεταφορικών και την συνολική χρέωση.

- 8 Ο μισθός του κυρίου **Ακρογιάλι** είναι 850€, ενώ σύμφωνα με το μισθολόγιο αυξάνεται κατά 3% ετησίως. Κάθε μήνα έχει αποφασίσει να αποταμιεύσει το 9% του μισθού για το όνειρό του που είναι η αγορά φουσκωτού σκάφους. Να αναπτύξετε αλγόριθμο που θα υπολογίζει και θα εκτυπώνει σε πόσους μήνες θα κατορθώσει να συγκεντρώσει το απαιτούμενο ποσό, ώστε να αγοράσει φουσκωτό αξίας 3,500€.
- 9 Ο κύριος **Αλγοριθμίδης** σύναψε δάνειο στην τράπεζα **Αεππούπολης** ώστε να ανακαινίσει το σπίτι του. Η τράπεζα του ανακοίνωσε το νέο της πρόγραμμα δανείων που είναι το εξής: Η πρώτη δόση είναι 100€, ενώ κάθε εξάμηνο αυξάνεται κατά 50€, μέχρι να φτάσει το ποσό των 400€ (η δόση δεν μπορεί να είναι μεγαλύτερη από 400€). Με τη συμπλήρωση του κάθε χρόνου από τη σύναψη του δανείου το υπολειπόμενο ποσό τοκίζεται με επιτόκιο 10.5%. Σημείωση: Κατά τη σύναψη του δανείου τοκίζεται το αρχικό ποσό. Να αναπτύξετε αλγόριθμο που θα διαβάσει το ποσό που σκέφτεται να δανειστεί ο κύριος **Αλγοριθμίδης** και θα εμφανίζει σε πόσους μήνες θα αποπληρώσει το δάνειο.
- 10 Το τυπογραφείο «**Copies**» εκτυπώνει διαφημιστικά φυλλάδια για κάθε ενδιαφερόμενο και χρησιμοποιεί δύο τιμολογιακές πολιτικές χρέωσης:
 Πάγια χρέωση 800€ και επιπλέον 0.75€ ανά φυλλάδιο.
 Χρέωση 3.20€ για καθένα από τα 300 πρώτα φυλλάδια, ενώ η τιμή για καθένα από τα επόμενα 200 είναι 2.90€. Τέλος, κάθε φυλλάδιο πέρα των 500 χρεώνεται με 2.30€
 Είναι προφανές ότι ο πρώτος ενδείκνυται στην περίπτωση που πρόκειται να εκτυπωθεί μεγάλος αριθμός φυλλαδίων, ενώ για λιγότερα φυλλάδια προτιμάται ο δεύτερος.
 Σημειώνεται επίσης, ότι τα φυλλάδια εκτυπώνονται ανά 100. Γνωστό ψητοπωλείο ενδιαφέρεται να εκτυπώσει διαφημιστικά φυλλάδια. Να αναπτύξετε αλγόριθμο που θα εντοπίζει και θα εμφανίζει το πλήθος των φυλλαδίων που πρέπει να παραγγείλει ώστε να είναι οικονομικότερος ο πρώτος τρόπος τιμολόγησης. Η διαδικασία που πρέπει να ακολουθήσετε είναι η εξής: να υπολογίζεται το κόστος των φυλλαδίων με τους δύο τρόπους ανά 100 φυλλάδια (100, 200, 300 ...) έως ότου το ποσό που προκύπτει με τη δεύτερη τιμολόγηση να είναι μεγαλύτερο από αυτό με την πρώτη.

- 11** Μια μεγάλη έκταση θα πουληθεί σε οικόπεδα. Να αναπτύξετε αλγόριθμο που θα διαβάσει τη συνολική έκταση και στη συνέχεια θα διαβάσει επαναληπτικά το μέγεθος κάθε οικοπέδου (που δεν μπορεί να είναι μικρότερο από το 15% της συνολικής έκτασης, ούτε μεγαλύτερο από τη διαθέσιμη έκταση). Μέχρι να υπάρχει οικόπεδο που δεν μπορεί να πουληθεί και θα εκτυπώνει πόσα οικόπεδα πουλήθηκαν, καθώς και το εμβαδόν της έκτασης που περίσσεψε για κοινόχρηστους χώρους.
- 12** Σύμφωνα με τον τελευταίο νόμο του Υπουργείου Οικονομικών, οι μεγαλοοφειλέτες μπορούν να προσέλθουν στις κατά τόπους εφορίες και να προβούν σε διακανονισμούς για την αποπληρωμή των οφειλών τους. Έτσι, αν κάποιος χρωστά περισσότερα από 100.000€, θα έχει έκπτωση 30%, ενώ διαφορετικά θα έχει έκπτωση 20%. Αν κάποιος πληρώσει μετρητοίς, τότε θα έχει επιπλέον έκπτωση 5% επί του ποσού της οφειλής. Να αναπτύξετε αλγόριθμο που θα διαβάσει επαναληπτικά το όνομα ενός οφειλέτη, το πλήθος της οφειλής του, καθώς και το πλήθος των δόσεων που θα εξοφλήσει (δόσεις = 1, σημαίνει πληρωμή μετρητοίς) και θα εκτυπώνει το ποσό κάθε δόσης στην εφορία. Η επανάληψη θα τερματίζεται όταν δοθεί ως όνομα το «τέλος» και ο αλγόριθμος θα επιστρέφει το συνολικό ποσό που θα εισρεύσει στα ταμεία του κράτους.
- 13** Να γράψετε πρόγραμμα που δίνει 3 ευκαιρίες για την ενεργοποίηση ενός κινητού τηλεφώνου με **PIN 2345**. Αν μια από τις 3 προσπάθειες είναι επιτυχημένη να εμφανίζει το μήνυμα **‘ΚΑΛΗΜΕΡΑ’**. Μετά την 3η ανεπιτυχή προσπάθεια να εμφανίζει το μήνυμα **‘ΕΙΣΑΓΕΤΕ ΤΟΝ ΚΩΔΙΚΟ ΡΥΚ’**.
- 14** Να αναπτύξετε αλγόριθμο που να διαβάσει έναν ακέραιο αριθμό N. Να διαβάσετε N αριθμούς και να εμφανίσετε το άθροισμά τους.
- 15** Να αναπτύξετε αλγόριθμο που να εκτυπώνει τους τριψήφιους θετικούς ακераίους των οποίων ένα ακριβώς ψηφίο είναι 3.
- 16** Το παραγοντικό ενός αριθμού δίνεται από την σχέση:
- $$N! = \begin{cases} \text{αόριστο} & \text{αν } N < 0 \\ 1 & \text{αν } N = 0 \\ 1 * 2 * 3 * \dots * N & \text{αν } N > 0 \end{cases}$$
- 17** Να αναπτύξετε αλγόριθμο που θα διαβάσει έναν αριθμό και θα εκτυπώνει το παραγοντικό του.
- 18** Η ημερήσια πίεση ενός ασθενή εκτιμάται ως ο μέσος όρος των 24 μετρήσεων που πραγματοποιούνται κατά τη διάρκεια μιας ημέρας. Να αναπτύξετε αλγόριθμο που θα διαβάσει για κάθε έναν από τους 35 ασθενείς μιας κλινικής το όνομα του και τις μετρήσεις για την ωριαία πίεσή του και θα εκτυπώνει το όνομά του και την ημερήσια πίεση του. Επίσης θα εκτυπώνονται το επί τοις εκατό ποσοστό των ασθενών με ημερήσια πίεση πάνω από 17.

- 19** Να αναπτύξετε αλγόριθμο που θα διαβάσει 2 αριθμούς και να ελέγχει να είναι θετικοί. Να εμφανίζει το ΜΚΔ.
- 20** Να γράψετε πρόγραμμα το οποίο θα υπολογίζει τον μικρότερο ακέραιο, ο οποίος είναι περιττός και 3. Να γραφεί πρόγραμμα που να υπολογίζει το άθροισμα $1*2 + 2*2 + \dots + n*2$.
- 21** Ο πληθωρισμός σε μία χώρα «τρέχει» με 17%. Αν ένας εργαζόμενος παίρνει αύξηση κάθε χρόνο 6% επί του μισθού του, υπολογίστε σε πόσα χρόνια η αγοραστική του δύναμη θα μειωθεί στο μισό. (Υπόδειξη: δεχθείτε ένα συγκεκριμένο ποσό σαν μισθό του εργαζομένου, πχ. 250000 δρχ. και γράψτε τον αλγόριθμο έτσι ώστε να υπολογίζει για κάθε χρόνο πόσο επηρεάζεται το ποσό αυτό από τον πληθωρισμό και από την αύξηση του μισθού. Στο τέλος του χρόνου που ψάχνετε θα πρέπει το υπολογιζόμενο ποσό να είναι ίσο ή μικρότερο από το μισό του αρχικού).
- 22** Σε μία ασφαλιστική εταιρεία δουλεύουν Α ασφαλιστές που ο καθένας τους μπορεί να έχει 1 έως Ν πελάτες. Να δοθεί πρόγραμμα που για κάθε ασφαλιστή να ζητά τον αριθμό των πελατών του και για κάθε πελάτη να ζητά το μέγεθος της ασφάλειας που έχει κάνει. Στο τέλος το πρόγραμμα να εμφανίζει τον αριθμό των ασφαλιστών, το συνολικό αριθμό των πελατών και ένα μήνυμα για το ποιος πελάτης έχει κάνει τη μεγαλύτερη ασφάλιση, π.χ. ως εξής: “Ο 2ος πελάτης του 4ου ασφαλιστή έχει κάνει τη μεγαλύτερη ασφάλεια ύψους 1.500 Euro.”
- 23** Ένας αγρότης, για να κάνει μία γεώτρηση στο κτήμα του, συμφώνησε τα εξής με τον ιδιοκτήτη του γεωτρύπανου. Το 1ο μέτρο θα κοστίσει 2000 δρχ. και αυξανόμενου του βάθους, θα αυξάνεται και η τιμή κάθε μέτρου κατά 500 EURO. Ο αγρότης διαθέτει 470.000 EURO. Να γίνει πρόγραμμα που να υπολογίζει το βάθος που μπορεί να πάει η γεώτρηση στο κτήμα.
- 24** Μια εταιρεία αμείβει του πωλητές της με μισθό 1300 EURO και ποσοστά επί των πωλήσεων σύμφωνα με τον παρακάτω πίνακα:

	ΠΩΛΗΣΕΙΣ	ΠΟΣΟΣΤΟ
1	Μέχρι 1.000	5%
2	Από 1.001 μέχρι 3.000	8%
3	Από 3.001 μέχρι 5.000	10%
4	Άνω των 5.001	12%

Αν ο πωλητής πουλήσει 4.000 θα πάρει για το 1° 1000 5%, για τα επόμενα 2000 ποσοστό 8% και για το 4° 10%

Να κατασκευάσετε πρόγραμμα που:

- Να διαβάζει τον κωδικό και τις πωλήσεις του πωλητή.
- Να υπολογίζει και να εκτυπώνει το συνολικό ποσό που δικαιούται για πριμ και το σύνολο της αμοιβής του.
- Η διαδικασία να επαναλαμβάνεται για άγνωστο αριθμό πωλητών μέχρι να εισαχθεί για κωδικός το 0.
- Να υπολογίζει το συνολικό ποσό που θα πληρώσει η εταιρεία για αμοιβές.
- Το μέσο όρο των πριμ.

**ΑΛΥΤΕΣ ΟΜΑΔΕΣ ΑΣΚΗΣΕΩΝ ΕΠΙ ΤΩΝ ΠΙΝΑΚΩΝ ΤΙΜΩΝ - ΛΟΓΙΚΩΝ ΕΚΦΡΑΣΕΩΝ – ΛΟΓΙΚΩΝ
ΔΙΑΓΡΑΜΜΑΤΩΝ**

ΠΙΝΑΚΕΣ ΤΙΜΩΝ

1. Τι θα εμφανίσει ο παρακάτω αλγόριθμος :

```

Αλγόριθμος Εμφάνιση
Για x από 2 μέχρι 20 με_βήμα 3
    Αν  $x < 8$  ΚΑΙ  $x < 17$  τότε
        Εμφάνισε x
    Τέλος_αν
Τέλος_επανάληψης
Τέλος Εμφάνιση
    
```

2. Πόσες φορές θα εμφανιστεί το μήνυμα “SOS” ,όταν εκτελεστεί το παρακάτω τμήμα αλγορίθμου:

```

ΓΙΑ λ ΑΠΟ 1 ΜΕΧΡΙ 4
    ΓΙΑ κ ΑΠΟ 1 ΜΕΧΡΙ λmod2 +1
        Εμφάνισε“SOS”
    ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ ΕΠΑΝΑΛΗΨΗΣ
    
```

3. Δίνεται το παρακάτω τμήμα αλγορίθμου :

```

 $x \leftarrow a$ 
 $y \leftarrow b$ 
Αρχή_επανάληψης
    Εμφάνισε x,y
     $x \leftarrow x + 1$ 
     $y \leftarrow y - x$ 
Μέχρις_ότου  $x > y$ 
    
```

Ποιες οι τιμές των μεταβλητών x και y που θα εμφανιστούν στην οθόνη του Η/Υ αν $a = 5$ και $b = 20$;

4. Δίνεται το παρακάτω τμήμα αλγορίθμου:

```

 $K \leftarrow 10$ 
ΟΣΟ  $K \geq 0$  ΕΠΑΝΕΛΑΒΕ
     $A \leftarrow K + 2$ 
    ΑΝ  $A < 10$  ΤΟΤΕ
         $I \leftarrow A + K$ 
    ΑΛΛΙΩΣ
         $I \leftarrow A - K$ 
    ΤΕΛΟΣ_ΑΝ
     $K \leftarrow K - 3$ 
    ΓΡΑΨΕ A, I
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    
```

- α) Ποιες οι τιμές των μεταβλητών A και I που θα εμφανιστούν με την εντολή ΓΡΑΨΕ A, I ;
β) Να μετατραπεί το παραπάνω τμήμα αλγορίθμου κάνοντας χρήση της εντολής ΓΙΑ...ΑΠΟ ..ΜΕΧΡΙ έτσι ώστε να εμφανίζονται οι ίδιες τιμές.

5. Έστω το παρακάτω τμήμα αλγορίθμου:

```
K ← 0  
ΓΙΑ I ΑΠΟ 0 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2  
    A ← I^2  
    K ← K+A  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
ΓΡΑΨΕ K
```

- α) Να γράψετε τον πίνακα τιμών για τις μεταβλητές I, K και A.
β) Να μετατρέψετε το παραπάνω τμήμα αλγορίθμου χρησιμοποιώντας τη δομή ΟΣΟΕΠΑΝΑΛΑΒΕ.

6. Μελετήστε προσεχτικά τα παρακάτω τμήματα προγράμματος. Αν τα δεδομένα είναι οι αριθμοί $\alpha=5$ και $\beta=4$ ποια είναι η εκτύπωση των παρακάτω τμημάτων προγράμματος καθώς και για $\alpha=2$, $\beta=3$.

(I)
 Διάβασε α, β
 Αν $\alpha \geq \beta$ τότε
 Αν $\beta > 0$ τότε
 $\alpha \leftarrow \alpha + 3$
 $\beta \leftarrow \beta - \alpha$
 αλλιώς
 $\beta \leftarrow \beta - 5$
 Τέλος_αν
 αλλιώς
 Αν $\alpha + 5 \geq \beta$ τότε
 $\beta \leftarrow 5 - \beta$
 αλλιώς
 $\alpha \leftarrow \alpha + \beta$
 Τέλος_αν
 Τέλος_αν
 Εκτύπωσε α, β

(I I)
 Διάβασε α, β
 Αν $\alpha \geq \beta$ και $\beta > 0$ τότε
 $\alpha \leftarrow \alpha + 3$
 $\beta \leftarrow \beta - \alpha$
 Τέλος_αν
 Αν $\alpha < \beta$ και $\alpha + 5 \geq \beta$ τότε
 $\beta \leftarrow 5 - \beta$
 Τέλος_αν
 $\beta \leftarrow \beta - 5$
 $\alpha \leftarrow \alpha + \beta$
 Εκτύπωσε α, β

7. Αν τα δεδομένα είναι οι αριθμοί $\alpha = 1, \beta = 4, \gamma = 3$ ποια είναι η εκτύπωση των παρακάτω τμημάτων προγράμματος καθώς και για $\alpha=1, \beta=2, \gamma=4$.

(I)
 Διάβασε α, β, γ
 Αν $\alpha = 1$ τότε
 Εκτύπωσε 'ΕΝΑ'
 Τέλος_αν
 Αν $\beta = 2$ τότε
 Εκτύπωσε 'ΔΥΟ'
 Τέλος_αν
 Αν $\gamma = 3$ τότε
 Εκτύπωσε 'ΤΡΙΑ'
 Τέλος_αν
 Εκτύπωσε 'ΚΑΝΕΝΑ'

(I I)
 Διάβασε α, β, γ
 Αν $\alpha = 1$ τότε
 Αν $\beta = 2$ τότε
 Αν $\gamma = 3$ τότε
 Εκτύπωσε 'ΤΡΙΑ'
 αλλιώς
 Εκτύπωσε 'ΔΥΟ'
 Τέλος_αν
 αλλιώς
 Εκτύπωσε 'ΕΝΑ'
 Τέλος_αν
 αλλιώς
 Εκτύπωσε 'ΚΑΝΕΝΑ'
 Τέλος_αν

8. Δίνεται ο παρακάτω αλγόριθμος. Ποιός είναι ο πίνακας τιμών;

Αλγόριθμος Πίνακας_Τιμών

$X \leftarrow 2$

$Y \leftarrow X^2 - 1$

$Z \leftarrow 2 * X + Y - 1$

Αν $(X > Y)$ τότε

$Y \leftarrow Z \bmod X$

$Z \leftarrow X^2$

Αλλιώς

$X \leftarrow Z \bmod Y$

$Z \leftarrow Y^2$

Τέλος_Αν

Εκτύπωσε X, Y, Z

Τέλος Πίνακας_Τιμών

9. Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί;
Να σχηματίσετε και το διάγραμμα ροής.

Αλγόριθμος Πίνακας_Τιμών

$\alpha \leftarrow 3$

$\beta \leftarrow 1$

$\gamma \leftarrow 5$

Αν $(\alpha \bmod 2 = 1)$ ή $(\beta \geq 2)$ τότε

$\gamma \leftarrow \gamma + 2$

Αν $(\gamma < \beta)$ τότε

$\alpha \leftarrow \alpha^3$

Αλλιώς

$\beta \leftarrow 4 * \beta$

Τέλος_Αν

Τέλος_Αν

$\alpha \leftarrow \alpha \bmod \beta$

$\beta \leftarrow \beta \bmod \gamma$

$\gamma \leftarrow \gamma \bmod \alpha$

Εκτύπωσε α, β, γ

Τέλος Πίνακας_Τιμών

10. Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου αν οι τιμές που θα δώσει ο χρήστης είναι $\alpha = 17$ και $\gamma = 4$. Τι θα εκτυπωθεί; Να σχηματίσετε και το διάγραμμα ροής.

Αλγόριθμος Πίνακας_Τιμών

Διάβασε α

Διάβασε γ

$\alpha \leftarrow \alpha + 1$

$\beta \leftarrow \alpha \bmod \gamma$

Αν $(\beta \geq 1)$ και $(\beta \leq 3)$ τότε

$\gamma \leftarrow \gamma + 2$

$\beta \leftarrow \gamma \text{ div } \beta$

Αν $(\gamma \leq \alpha)$ τότε

$\alpha \leftarrow \alpha + 3$

Αλλιώς

$\alpha \leftarrow \alpha / 2$

Τέλος_Αν

Εκτύπωσε α, β, γ

Τέλος_Αν

$\alpha \leftarrow \alpha + \beta$

$\beta \leftarrow \beta * \gamma$

$\gamma \leftarrow \gamma - \alpha$

Εκτύπωσε α, β, γ

Τέλος Πίνακας_Τιμών

11. Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου αν οι τιμές που θα δώσει ο χρήστης είναι $\alpha = 3$ και $\gamma = 7$. Τι θα εκτυπωθεί;

Αλγόριθμος Πίνακας_Τιμών

Διάβασε α

Διάβασε γ

$\beta \leftarrow 2 * \alpha + \gamma$

$\delta \leftarrow \beta \bmod \gamma$

Αν $(\alpha \leq \beta)$ τότε

$\gamma \leftarrow \gamma + 2$

Αν $(\gamma \leq \alpha)$ τότε

$\alpha \leftarrow 3 * \beta$

$\beta \leftarrow \delta + 3$

Αλλιώς

$\alpha \leftarrow \alpha \text{ div } 2 - 5$

Τέλος_Αν

Αλλιώς

$\gamma \leftarrow 3 * \gamma - \delta$

$\beta \leftarrow \beta \text{ div } 2 + \delta$

Τέλος_Αν

Εκτύπωσε $\alpha, \beta, \gamma, \delta$

$\alpha \leftarrow \alpha + \beta$

$\gamma \leftarrow \gamma * \alpha$

Αν $(\alpha \geq \delta)$ τότε

$\alpha \leftarrow \alpha - \delta + \beta$

$\beta \leftarrow \beta + \delta$

$\gamma \leftarrow \gamma \text{ div } 3$

Αλλιώς

$\alpha \leftarrow \gamma - \alpha - 2 * \beta$

$\beta \leftarrow \beta + 6$

$\gamma \leftarrow (\gamma - 1) / 4$

Τέλος_Αν

Εκτύπωσε α, β, γ

Τέλος Πίνακας_Τιμών

12. Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί;

```

Αλγόριθμος Πίνακας_Τιμών
α ← 3
β ← 31
γ ← (β - α) / 2
δ ← (α + β) div 3
Αν (α <= β) τότε
    γ ← γ - 3
    Αν (γ <= α) τότε
        β ← β - δ + 3
        α ← 3 * β
    Αλλιώς
        α ← α div 2 + 2
Τέλος_Αν
Αλλιώς
    β ← β div 2 + δ
    γ ← 3 * γ - δ
Τέλος_Αν
Εκτύπωσε α, β, γ, δ
α ← α + β
γ ← γ * δ
Αν γ < 20 τότε
    α ← α - δ + β
    β ← β + δ
    γ ← γ * 3
Αλλιώς_αν γ <= 80 τότε
    α ← α + β ^ 2
    β ← (β + δ ^ 2) div 2
    γ ← γ ^ 3
Αλλιώς
    α ← (α - δ) mod 7
    β ← β - δ
    γ ← γ div α
Τέλος_Αν
δ ← δ - (α + β) + γ
Εκτύπωσε α, β, γ, δ
Τέλος Πίνακας_Τιμών
    
```

13. Πόσες φορές θα εκτελεστεί η παρακάτω επανάληψη;

```

Κ ← 0
Μ ← 10
Όσο Μ > 0 επανάλαβε
    Κ ← Κ + 1
    Μ ← Μ - 1
Τέλος_επανάληψης
α) 0      β) 1      γ) 10      δ) Τίποτα από τα παραπάνω
    
```

14. Δίνεται ο παρακάτω αλγόριθμος. Πόσες φορές θα εκτελεστεί;

```

X ← 0
Όσο X <> 1 επανάλαβε
    Για I από 1 μέχρι 10
        X ← X - 1
    Τέλος_επανάληψης
Τέλος_επανάληψης

```

α) 0 β) άπειρες γ) 1 δ) Τίποτα από τα παραπάνω

15. Ποιό θα είναι το αποτέλεσμα εκτέλεσης των παρακάτω εντολών;

```

Για I από 1 μέχρι 15
    Εμφάνισε I*2 - 1
Τέλος_επανάληψης

```

A) Εμφανίζει την έκφραση $I*2-1$.
 B) Εμφανίζει όλους τους περιττούς αριθμούς από το 1 έως το 15.
 Γ) Εμφανίζει όλους τους ακεραίους αριθμούς από το 1 έως το 15.
 Δ) Τίποτα από τα παραπάνω.

16. Να δοθούν οι τιμές όλων των μεταβλητών κατά τη διάρκεια εκτέλεσης του παρακάτω αλγορίθμου.

```

Αλγόριθμος Ένα
A ← 2
B ← 1
Για I από 3 μέχρι 8 με_βήμα 2
    A ← A + I
    B ← B + A
    K ← A + B
    Αν A < B - 3 τότε
        K ← K + 1
        Αν K > A + B
            A ← B
            B ← A
        Αλλιώς
            A ← K
            K ← B
            B ← K
    Τέλος_αν
Αλλιώς
    Αν B >= K + 1 τότε
        A ← A + 1
    Τέλος_αν
    Αν A <= B - 3 τότε
        K ← K - 1

```

```

        Τέλος_αν
    Τέλος_αν
    Αν  $A = B$  τότε
         $A \leftarrow A/2$ 
    Τέλος_αν
     $B \leftarrow K - A$ 
    Εμφάνισε A, B, K, I
Τέλος_επανάληψης
Εμφάνισε A, B, K, I
Τέλος Ένα
    
```

Τι θα εμφανίσει ο αλγόριθμος; Να σχηματίσετε και το διάγραμμα ροής.

17. Κάντε το ίδιο για το παρακάτω αλγοριθμικό κομμάτι.

```

     $A \leftarrow 2$ 
    Για I από 5 μέχρι 2 με_βήμα -1
         $A \leftarrow A + 1$ 
         $B \leftarrow I + A$ 
        Εμφάνισε A, B, I
    Τέλος_επανάληψης
    Εμφάνισε A, B, I
    Αν  $B \leq A + I$  τότε
         $M \leftarrow A + B/2$ 
    Αλλιώς
         $M \leftarrow A - B/2$ 
        Όσο  $M \geq -4$  επανάλαβε
             $A \leftarrow A + M$ 
             $M \leftarrow M - 2$ 
            Εμφάνισε A, M
        Τέλος_επανάληψης
    Τέλος_αν
    Εμφάνισε A, B, I, M
    
```

18. Τι θα εμφανιστεί αν εκτελεστεί το παρακάτω αλγοριθμικό απόσπασμα;
Να σχηματίσετε και το διάγραμμα ροής.

```

     $X \leftarrow 1$ 
    Όσο  $X < 5$  επανάλαβε
         $A \leftarrow X + 2$ 
         $B \leftarrow 3 * A - 4$ 
         $\Psi \leftarrow B - A + 4$ 
        Αν  $A > B$  τότε
            Αν  $A > \Psi$  τότε
                 $MAX \leftarrow A$ 
            Αλλιώς
                 $MAX \leftarrow \Psi$ 
        Τέλος_αν
    
```

```

    Αν B>Ψ τότε
        MAX ← B
    Αλλιώς
        MAX ← Ψ
    Τέλος_αν
Τέλος_αν
Εμφάνισε X,A,B,Ψ,MAX
X ← X+2
Τέλος_επανάληψης

```

- 19.** Ποιές οι τιμές των μεταβλητών X,M,Z σε όλες τις επαναλήψεις;
Να σχηματίσετε και το διάγραμμα ροής.

```

M ← 0
Z ← 0
Για X από 0 μέχρι 10 με_βήμα 2
    Αν X<5 τότε
        Z ← Z+X
    Αλλιώς
        M ← M+X-1
    Τέλος_αν
Τέλος_επανάληψης.

```

- 20.** Τι θα εμφανιστεί με την εκτέλεση του παρακάτω αλγοριθμικού αποσπάσματος;

```

K ← 24
Λ ← 40
X ← K
Y ← Λ
Αν X<Y τότε
    TEMP ← X
    X ← Y
    Y ← TEMP
Τέλος_αν
Όσο Y<>0 επανάλαβε
    TEMP ← Y
    Y ← X MOD Y
    X ← TEMP
Εμφάνισε X,Y
Τέλος_επανάληψης
Y ← (K*Λ) DIV X
Εμφάνισε X,Y

```

21. Τι τιμές έχουν οι μεταβλητές μετά το τέλος του παρακάτω τμήματος αλγορίθμου για είσοδο α) -1 β) 7;

```

Διάβασε X
Y ← -2+(X+1)/2
Όσο X ≤ Y ή Y ≥ 0 επανάλαβε
    Αν X > Y τότε
        Y ← Y-1
        X ← X-2
    Αλλιώς
        Y ← Y-2
        X ← X+1
Τέλος_αν
Τέλος_επανάληψης
Να σχηματίσετε και το διάγραμμα ροής.

```

22. Ποιά από τα παρακάτω τμήματα δίνουν το ίδιο αποτέλεσμα όταν εκτελεστούν;

A) $X \leftarrow 2$ $Y \leftarrow 3$ Όσο $Y > 0$ επανάλαβε $Y \leftarrow Y-1$ $X \leftarrow 2+X$ Τέλος_επανάληψης Εμφάνισε X	B) $X \leftarrow 2$ $Y \leftarrow 11$ Όσο $Y > 8$ επανάλαβε $Y \leftarrow Y-1$ $X \leftarrow 2+X$ Τέλος_επανάληψης Εμφάνισε X	Γ) $X \leftarrow 2$ $Y \leftarrow 3$ Όσο $Y > 4$ επανάλαβε $Y \leftarrow Y-1$ $X \leftarrow 3+X$ Τέλος_επανάληψης Εμφάνισε X
Δ) $X \leftarrow 2$ $Y \leftarrow 2$ Όσο $Y \geq 0$ επανάλαβε $Y \leftarrow Y-1$ $X \leftarrow 2+X$ Τέλος_επανάληψης Εμφάνισε X	Ε) $X \leftarrow 3$ $Y \leftarrow 6$ Όσο $Y > 0$ επανάλαβε $Y \leftarrow Y-2$ $X \leftarrow 1+X$ Τέλος_επανάληψης Εμφάνισε X	Ζ) $X \leftarrow 2$ $Y \leftarrow -3$ Όσο $Y \leq -1$ επανάλαβε $Y \leftarrow Y+1$ $X \leftarrow 2+X$ Τέλος_επανάληψης Εμφάνισε X

23. Ποιά από τα παρακάτω τμήματα δίνουν το ίδιο αποτέλεσμα όταν εκτελεστούν;

A) $X \leftarrow 2$ Για Y από 1 μέχρι 5 $X \leftarrow X + 2$ Τέλος_επανάληψης Εμφάνισε X	B) $X \leftarrow 2$ Για Y από -1 μέχρι 4 $X \leftarrow X + 2$ Τέλος_επανάληψης Εμφάνισε X
Γ) $X \leftarrow 2$ Για Y από -5 μέχρι 5 με_βήμα 2 $X \leftarrow X + 2$	Δ) $X \leftarrow 2$ Για Y από 1 μέχρι 12 με_βήμα 2 $X \leftarrow X + 2$

Τέλος_επανάληψης
Εμφάνισε X

Τέλος_επανάληψης
Εμφάνισε X

24. Ποιά από τα παρακάτω τμήματα δίνουν το ίδιο αποτέλεσμα όταν εκτελεστούν;

A)

$X \leftarrow 2$
 $Y \leftarrow -3$
 Όσο $Y \leq 0$ επανάλαβε
 $Y \leftarrow Y+1$
 $X \leftarrow Y+X$
 Τέλος_επανάληψης
 Εμφάνισε X

B)

$X \leftarrow 2$
 Για Y από -3 μέχρι 1
 $X \leftarrow X + Y$
 Τέλος_επανάληψης
 Εμφάνισε X

Γ)

$X \leftarrow 2$
 $Y \leftarrow -3$
 Όσο $Y \leq 0$ επανάλαβε
 $X \leftarrow Y+X$
 $Y \leftarrow Y+1$
 Τέλος_επανάληψης
 Εμφάνισε X

Δ)

$X \leftarrow 2$
 $Y \leftarrow -3$
 Όσο $Y \leq 0$ επανάλαβε
 $X \leftarrow Y+X+1$
 $Y \leftarrow Y+1$
 Τέλος_επανάληψης
 Εμφάνισε X

Ε)

$X \leftarrow 2$
 Για Y από -3 μέχρι 0
 $X \leftarrow X + Y$
 Τέλος_επανάληψης
 Εμφάνισε X

Ζ)

$X \leftarrow 2$
 $Y \leftarrow -3$
 Όσο $Y \leq 0$ επανάλαβε
 $Y \leftarrow Y+1$
 $X \leftarrow Y+X-1$
 Τέλος_επανάληψης
 Εμφάνισε X

25. Θεωρήστε ότι $A=8$, $B=5$, $\Gamma=4$ και $\Delta=1$. Τι θα εμφανίσουν τα παρακάτω τμήματα αλγορίθμου;

A)

Για Y από 2 μέχρι 14 με_βήμα Γ
 Εμφάνισε Y
 Τέλος_επανάληψης
 Εμφάνισε X

B)

$K \leftarrow 0$
 $\Lambda \leftarrow 0$
 Για Y από 3 μέχρι 6
 $K \leftarrow K + \Gamma - \Delta$
 $\Lambda \leftarrow \Lambda * B$
 Τέλος_επανάληψης
 Εμφάνισε K, Λ

Γ)

$X \leftarrow 2$
 Για Y από 1 μέχρι A με_βήμα 2
 $X \leftarrow X * Y$
 Εμφάνισε Y
 Τέλος_επανάληψης
 Εμφάνισε X

Δ)

$X \leftarrow 5$
 Για Y από Γ μέχρι A
 $X \leftarrow X + \Delta$
 Εμφάνισε X^2
 Τέλος_επανάληψης

- 26.** Πόσα αστεράκια θα εκτυπωθούν κατά την εκτέλεση του παρακάτω αλγορίθμου;
Να σχηματίσετε και το διάγραμμα ροής.

```

Αλγόριθμος Αστεράκια
  Για i από 8 μέχρι 2 με_βήμα -2
    Για j από 2 μέχρι 7 με_βήμα 2
      Αν i > j τότε
        Εκτύπωσε '*'
      Τέλος_Αν
    Τέλος_Επανάληψης
  Τέλος_Επανάληψης
Τέλος Αστεράκια

```

- 27.** Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί ;
Να σχηματίσετε και το διάγραμμα ροής.

```

Αλγόριθμος Άσκηση
  α ← 0
  Όσο (α <= 22) επανάλαβε
    Για i από 1 μέχρι 3
      α ← α + i
    Τέλος_Επανάληψης
    α ← α + 5
    Εκτύπωσε α
  Τέλος_Επανάληψης
  Εκτύπωσε α
Τέλος Άσκηση

```

- 28.** Να σχηματίσετε τον πίνακα τιμών του παρακάτω αλγορίθμου. Τι θα εκτυπωθεί ;
Να σχηματίσετε και το διάγραμμα ροής.

```

Αλγόριθμος Πίνακας_Τιμών
  β ← 0
  Για i από 1 μέχρι 3
    α ← 2 * i
    Για j από i μέχρι α
      β ← β + i * j
    Τέλος_Επανάληψης
    Εμφάνισε β
  Τέλος_Επανάληψης
Τέλος Πίνακας_Τιμών

```

- 29.** Να σχηματίσετε τον πίνακα τιμών του παρακάτω τμήματος αλγορίθμου. Τι θα εκτυπωθεί ;
Να σχηματίσετε και το διάγραμμα ροής.

```

β ← 3
τ ← 0
Για i από 1 μέχρι 4
  β ← β + 2

```

Για j από β μέχρι i με_βήμα -2
 $\tau \leftarrow \tau + j$
 Τέλος_Επανάληψης
 Εμφάνισε τ, β
 Τέλος_Επανάληψης

30. Δίνεται το παρακάτω τμήμα αλγορίθμου στον οποίο έχουν αριθμηθεί οι εκτελέσιμες εντολές.

1. $X \leftarrow 2$
2. $Y \leftarrow 0$
3. $I \leftarrow 7$
4. Όσο $I \geq 4$ επανάλαβε
5. $X \leftarrow X + 2$
6. Αν $(X > 4)$ τότε
 - i. $X \leftarrow X + I \text{ div } Y$
7. Αλλιώς
 - i. $Y \leftarrow Y + 2$
8. Τέλος_Αν
9. Εκτύπωσε X, Y, I
10. $I \leftarrow I - 2$
11. Τέλος_επανάληψης
12. $X \leftarrow Y + I \text{ div } X$
13. $X \leftarrow X - Y$
14. Εκτύπωσε X, Y, I

Επίσης δίνεται το ακόλουθο υπόδειγμα πίνακα τιμών:

Αριθμός Γραμμής	Συνθήκη	Έξοδος	X	Y	I

Στη στήλη με τίτλο «αριθμός γραμμής» καταγράφεται ο αριθμός γραμμής της εντολής που εκτελείται. Στη στήλη με τίτλο «συνθήκη» καταγράφεται η λογική τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ, εφόσον η εντολή που εκτελείται περιλαμβάνει συνθήκη.

Στη στήλη με τίτλο «έξοδος» καταγράφεται η τιμή εξόδου, εφόσον η εντολή που εκτελείται είναι εντολή εξόδου.

Στη συνέχεια του πίνακα υπάρχει μια στήλη για κάθε μεταβλητή του αλγορίθμου. Να μεταφέρετε τον πίνακα στο τετράδιό σας και να τον συμπληρώσετε εκτελώντας τις εντολές του τμήματος αλγορίθμου ως εξής: Για κάθε εντολή που εκτελείται να γράψετε σε νέα γραμμή του πίνακα τον αριθμό της γραμμής της και το αποτέλεσμα της στην αντίστοιχη στήλη.

ΑΣΚΗΣΕΙΣ ΜΕ ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

1. Να διατυπώσετε σε λογικές εκφράσεις τις παρακάτω προτάσεις.

- i. Το α ανήκει στο διάστημα $[-5, 6)$
- ii. Το α είναι μικρότερο του 3 ή μεγαλύτερο του 15
- iii. Το α είναι ίσο με το β και το γ
- iv. Το α δεν έχει την τιμή 3
- v. Το α είναι μικρότερο του 2 ή το β είναι μεγαλύτερο του 78
- vi. α και β αληθή και γ ψευδές
- vii. το α αληθές και ένα από τα β, γ αληθές

2. Ποιό είναι το λογικό αποτέλεσμα (αληθής ή ψευδής) από την εκτέλεση των παρακάτω πράξεων αν οι εξής μεταβλητές έχουν τιμές:

A = 10, B = 2, Γ = -4, Δ = 9 και E = 1

- I. $(A > B) \vee (\Delta = 10)$
- II. $(\Delta \geq B) \wedge (E \neq \Gamma)$
- III. $\neg (E \leq \Gamma) \vee (\Delta \leq \Gamma)$
- IV. $\neg (B \leq \Gamma) \wedge (\Delta < 2)$
- V. $\neg (O \vee (B \leq E) \vee (E \leq B))$
- VI. $((E \leq A) \wedge (E \geq \Gamma)) \wedge \neg (\Gamma \geq A)$
- VII. $\neg (O \vee (A \geq 2) \wedge (\Gamma \neq 9))$

3. Η παράσταση $x - (x \text{ div } y) * y$ ισοδυναμεί με την πράξη:

- α) $x \text{ div } y$ β) $x \text{ mod } y$ γ) $x = y$ δ) Τίποτα από τα παραπάνω

4. Να τροποποιήσετε τις παρακάτω λογικές συνθήκες έτσι ώστε να δίνουν το ίδιο αποτέλεσμα για οποιαδήποτε τιμή των μεταβλητών x, y, w χρησιμοποιώντας διάζευξη (ή) αντί για σύζευξη (και).

- α) $x \leq y$ και $y \text{ mod } 2 = w$
 β) $y > w$ και $x \neq y + w$

5. Ποιό είναι το αποτέλεσμα από την εκτέλεση των παρακάτω πράξεων.

- i. $14 \text{ mod } 5 - 25 \text{ mod } 8$
- ii. $3 * (3 \text{ mod } 2) + 4 \text{ div } (5 \text{ mod } 3)$
- iii. $13 \text{ mod } (27 \text{ div } 4)$
- iv. $2^3 + 3 * (27 \text{ mod } (25 \text{ mod } 7))$
- v. $13/2 - 3 \text{ mod } 2 - 3 \text{ div } 2$

6. Να συμπληρώσετε τους παρακάτω πίνακες αλήθειας των δυο προτάσεων A, B και των τριών λογικών πράξεων.

A	B	A ή B	A και B	Όχι A

7. Αν $X = 15$, $Y = -3$ και $Z = 2$, να χαρακτηρίσετε τις ακόλουθες εκφράσεις χρησιμοποιώντας μια από τις λέξεις Αληθής ή Ψευδής.

- A) $X > Z$
 B) ΟΧΙ ($X + Y > 8$)
 Γ) ($X > Y$) ΚΑΙ ($Z < 3$)
 Δ) ($X > 10$) Η ($Y > 2$) ΚΑΙ ($Z > Y$)

8. Δίνονται οι τιμές των μεταβλητών $A = 5$, $B = 7$ και $\Gamma = -3$. Να χαρακτηρίσετε κάθε έκφραση που ακολουθεί με το γράμμα Α αν είναι αληθής ή με το γράμμα Ψ αν είναι ψευδής.

- A) ΟΧΙ ($A + B < 10$)
 B) ($A \geq B$) Η ($\Gamma < B$)
 Γ) ($(A > B)$ ΚΑΙ ($\Gamma < A$)) Η ($\Gamma > 5$)
 Δ) (ΟΧΙ ($A < B$)) ΚΑΙ ($B + \Gamma < 2 * A$)

9. Αν $\alpha = 5$, $\beta = 7$ και $\gamma = 10$, να χαρακτηρίσετε τις παρακάτω προτάσεις χρησιμοποιώντας μια από τις λέξεις Ψευδής ή Αληθής.

- Πρόταση Α: (όχι ($\alpha + 2 \geq \beta$)) ή $\beta + 3 = \gamma$
 Πρόταση Β: $\alpha + 2 * \beta < 20$ και $2 * \alpha = \gamma$

10. Αν $X = 3$, $Y = -2$ και $Z = -1$ να χαρακτηρίσετε τις παρακάτω προτάσεις χρησιμοποιώντας μια από τις λέξεις Αληθής ή Ψευδής.

- Πρόταση Α: $(X + Y) * Z > 0$
 Πρόταση Β: $(X - Y) * Z = -5$
 Πρόταση Γ: $X * Z > 0$
 Πρόταση Δ: $Z > Y$

ΑΣΚΗΣΕΙΣ ΜΕ ΛΟΓΙΚΑ ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΙ ΜΕΤΑΤΡΟΠΕΣ

1. Έστω το παρακάτω τμήμα αλγορίθμου:

```

K ← 0
ΓΙΑ Ι ΑΠΟ 0 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ 2
    A ← I^2
    K ← K+A
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Κ
    
```

α) Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα)

β) Να μετατρέψετε το παραπάνω τμήμα αλγορίθμου σε Διάγραμμα Ροής .

(Λογικό Διάγραμμα) χρησιμοποιώντας αποκλειστικά τη δομή

I. ΟΣΟΕΠΑΝΑΛΑΒΕ.

II. ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ..... ΜΕΧΡΙΣ_ΟΤΟΥ.

2. Έστω τα παρακάτω τμήματα αλγορίθμου:

```

(I)
Διάβασε α, β
Αν α >= β τότε
    Αν β > 0 τότε
        α ← α + 3
        β ← β - α
    αλλιώς
        β ← β - 5
Τέλος_αν
αλλιώς
    Αν α + 5 >= β τότε
        β ← 5 - β
    αλλιώς
        α ← α + β
Τέλος_αν
    
```

Τέλος_αν
Εκτύπωσε α,β

Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) για τα (I) , (II)

```

(II)
Διάβασε α, β
Αν α >= β και β > 0 τότε
    α ← α + 3
    β ← β - α
Τέλος_αν
Αν α < β και α + 5 >= β τότε
    β ← 5 - β
Τέλος_αν
β ← β - 5
α ← α + β
Εκτύπωσε α,β
    
```

3. Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) του παρακάτω αλγορίθμου.

```

Αλγόριθμος ΛΔ
α ← 3
β ← 1
γ ← 5
Αν (α mod 2 = 1) ή (β >= 2) τότε
    γ ← γ + 2
Αν (γ < β) τότε
    α ← α ^ 3
    
```

Αλλιώς

$\beta \leftarrow 4 * \beta$

Τέλος_Αν

Τέλος_Αν

$\alpha \leftarrow \alpha \bmod \beta$

$\beta \leftarrow \beta \bmod \gamma$

$\gamma \leftarrow \gamma \bmod \alpha$

Εκτύπωσε α, β, γ

Τέλος_ΛΔ

4. Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) του παρακάτω τμήματος αλγορίθμου.

Διάβασε α

Διάβασε γ

$\alpha \leftarrow \alpha + 1$

$\beta \leftarrow \alpha \bmod \gamma$

Αν $(\beta \geq 1)$ και $(\beta \leq 3)$ τότε

$\gamma \leftarrow \gamma + 2$

$\beta \leftarrow \gamma \div \beta$

Αν $(\gamma \leq \alpha)$ τότε

$\alpha \leftarrow \alpha + 3$

Αλλιώς

$\alpha \leftarrow \alpha / 2$

Τέλος_Αν

Εκτύπωσε α, β, γ

Τέλος_Αν

$\alpha \leftarrow \alpha + \beta$

$\beta \leftarrow \beta * \gamma$

$\gamma \leftarrow \gamma - \alpha$

Εκτύπωσε α, β, γ

5. Να σχηματίσετε τον πίνακα τιμών του παρακάτω τμήματος αλγορίθμου αν οι τιμές που θα δώσει ο χρήστης είναι $\alpha = 3$ και $\gamma = 7$. Τι θα εκτυπωθεί;

Διάβασε α

Διάβασε γ

$\beta \leftarrow 2 * \alpha + \gamma$

$\delta \leftarrow \beta \bmod \gamma$

Αν $(\alpha \leq \beta)$ τότε

$\gamma \leftarrow \gamma + 2$

Αν $(\gamma \leq \alpha)$ τότε

$\alpha \leftarrow 3 * \beta$

$\beta \leftarrow \delta + 3$

Αλλιώς

$\alpha \leftarrow \alpha \div 2 - 5$

Τέλος_Αν

Αλλιώς

```

 $\gamma \leftarrow 3 * \gamma - \delta$ 
 $\beta \leftarrow \beta \text{ div } 2 + \delta$ 
Τέλος_Αν
Εκτύπωσε  $\alpha, \beta, \gamma, \delta$ 
 $\alpha \leftarrow \alpha + \beta$ 
 $\gamma \leftarrow \gamma * \alpha$ 
Αν  $(\alpha \geq \delta)$  τότε
     $\alpha \leftarrow \alpha - \delta + \beta$ 
     $\beta \leftarrow \beta + \delta$ 
     $\gamma \leftarrow \gamma \text{ div } 3$ 
Αλλιώς
     $\alpha \leftarrow \gamma - \alpha - 2 * \beta$ 
     $\beta \leftarrow \beta + 6$ 
     $\gamma \leftarrow (\gamma - 1) / 4$ 
Τέλος_Αν
Εκτύπωσε  $\alpha, \beta, \gamma$ 

```

6. Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) του παρακάτω τμήματος αλγορίθμου.

```

 $A \leftarrow 2$ 
 $B \leftarrow 1$ 
Για  $I$  από 3 μέχρι 8 με_βήμα 2
     $A \leftarrow A + I$ 
     $B \leftarrow B + A$ 
     $K \leftarrow A + B$ 
    Αν  $A < B - 3$  τότε
         $K \leftarrow K + 1$ 
        Αν  $K > A + B$ 
             $A \leftarrow B$ 
             $B \leftarrow A$ 
        Αλλιώς
             $A \leftarrow K$ 
             $K \leftarrow B$ 
             $B \leftarrow K$ 
        Τέλος_αν
    Αλλιώς
        Αν  $B \geq K + 1$  τότε
             $A \leftarrow A + 1$ 
        Τέλος_αν
        Αν  $A \leq B - 3$  τότε
             $K \leftarrow K - 1$ 
        Τέλος_αν
    Τέλος_αν
    Αν  $A = B$  τότε
         $A \leftarrow A / 2$ 
    Τέλος_αν
     $B \leftarrow K - A$ 

```

Εμφάνισε A, B, K, I

Τέλος_επανάληψης

Εμφάνισε A, B, K, I

Να μετατρέψετε το παραπάνω τμήμα αλγορίθμου σε Διάγραμμα Ροής (Λογικό Διάγραμμα) χρησιμοποιώντας αποκλειστικά τη δομή.

I. ΟΣΟΕΠΑΝΑΛΑΒΕ.

II. ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ..... ΜΕΧΡΙΣ_ΟΤΟΥ.

7. Κάντε το ίδιο για το παρακάτω αλγοριθμικό κομμάτι.

$A \leftarrow 2$

Για I από 5 μέχρι 2 με_βήμα -1

$A \leftarrow A + 1$

$B \leftarrow I + A$

Εμφάνισε A, B, I

Τέλος_επανάληψης

Εμφάνισε A, B, I

Αν $B \leq A + I$ τότε

$M \leftarrow A + B/2$

Αλλιώς

$M \leftarrow A - B/2$

Όσο $M \geq -4$ επανάλαβε

$A \leftarrow A + M$

$M \leftarrow M - 2$

Εμφάνισε A, M

Τέλος_επανάληψης

Τέλος_αν

Εμφάνισε A, B, I, M

8. Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) του παρακάτω τμήματος αλγορίθμου.

$X \leftarrow 1$

Όσο $X < 5$ επανάλαβε

$A \leftarrow X + 2$

$B \leftarrow 3 * A - 4$

$\Psi \leftarrow B - A + 4$

Αν $A > B$ τότε

Αν $A > \Psi$ τότε

$MAX \leftarrow A$

Αλλιώς

$MAX \leftarrow \Psi$

Τέλος_αν

Αλλιώς

Αν $B > \Psi$ τότε

$MAX \leftarrow B$

Αλλιώς

$MAX \leftarrow \Psi$

Τέλος_αν

Τέλος_αν
Εμφάνισε X,A,B,Ψ,MAX
 $X \leftarrow X+2$
Τέλος_επανάληψης

Να μετατρέψετε το παραπάνω τμήμα αλγορίθμου σε Διάγραμμα Ροής (Λογικό Διάγραμμα) χρησιμοποιώντας αποκλειστικά τη δομή.

- I. ΓΙΑ .
- II. ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ..... ΜΕΧΡΙΣ_ΟΤΟΥ.

9. Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) του παρακάτω τμήματος αλγορίθμου.

$M \leftarrow 0$
 $Z \leftarrow 0$
Για X από 0 μέχρι 10 με_βήμα 2
 Αν $X < 5$ τότε
 $Z \leftarrow Z+X$
 Αλλιώς
 $M \leftarrow M+X-1$
Τέλος_αν
Τέλος_επανάληψης.

Να μετατρέψετε το παραπάνω τμήμα αλγορίθμου σε Διάγραμμα Ροής (Λογικό Διάγραμμα) χρησιμοποιώντας αποκλειστικά τη δομή

- I. ΟΣΟΕΠΑΝΑΛΑΒΕ.
- II. ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ..... ΜΕΧΡΙΣ_ΟΤΟΥ.

10. Να σχεδιαστεί Διάγραμμα Ροής (Λογικό Διάγραμμα) του παρακάτω τμήματος αλγορίθμου.

$K \leftarrow 24$
 $\Lambda \leftarrow 40$
 $X \leftarrow K$
 $Y \leftarrow \Lambda$
Αν $X < Y$ τότε
 $TEMP \leftarrow X$
 $X \leftarrow Y$
 $Y \leftarrow TEMP$
Τέλος_αν
Όσο $Y < > 0$ επανάλαβε
 $TEMP \leftarrow Y$
 $Y \leftarrow X \text{ MOD } Y$
 $X \leftarrow TEMP$
 Εμφάνισε X,Y
Τέλος_επανάληψης
 $Y \leftarrow (K * \Lambda) \text{ DIV } X$
Εμφάνισε X,Y

Να μετατρέψετε το παραπάνω τμήμα αλγορίθμου σε Διάγραμμα Ροής (Λογικό Διάγραμμα) χρησιμοποιώντας αποκλειστικά τη δομή

- I. ΓΙΑ .
- II. ΑΡΧΗ ΕΠΑΝΑΛΗΨΗΣ..... ΜΕΧΡΙΣ_ΟΤΟΥ.

EXTRA ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ

ΕΠΙ ΠΙΝΑΚΩΝ ΤΙΜΩΝ ΚΑΙ ΤΩΝ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ

1. Να μετατραπεί παρακάτω τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής.

```
Διάβασε x
Αν  $x \leq 0$  τότε
    Διάβασε y
    Αν  $y - x < 20$  τότε
         $y \leftarrow y + 10$ 
         $x \leftarrow x + 2$ 
    Τέλος_αν
    Εμφάνισε y
Αλλιώς
     $x \leftarrow x - 5$ 
Τέλος_αν
Εμφάνισε x
```

2. α) Να μετατραπεί παρακάτω τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής.

```
Διάβασε x
Αν  $x < 50$  τότε
     $x \leftarrow x - x * 10/100$ 
Αλλιώς_αν  $x < 150$  τότε
     $x \leftarrow x - x * 20/100$ 
Αλλιώς_αν  $x < 250$  τότε
     $x \leftarrow x - x * 30/100$ 
Αλλιώς
     $x \leftarrow x - x * 40/100$ 
Τέλος_αν
Εμφάνισε x
```

β) Να μετατραπεί το παραπάνω τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής κάνοντας αποκλειστική χρήση των απλών δομών επιλογής.

γ) Να μετατραπεί το παραπάνω τμήμα αλγορίθμου κάνοντας αποκλειστική χρήση δομής ΕΠΙΛΕΞΕ.

3. Να μετατραπεί το τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής.

Διάβασε x

$y \leftarrow 0$

Για i από 1 μέχρι 8 με_βήμα 2

Εμφάνισε x

$x \leftarrow x + 1$

$y \leftarrow y + x$

Τέλος_επανάληψης

Εμφάνισε y

β) Να μετατραπεί ο ακόλουθος αλγόριθμος σε ισοδύναμο διάγραμμα ροής κάνοντας αποκλειστική χρήση των δομών επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ αντί της ΓΙΑ.

4. α) Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής.

$x \leftarrow 6$

$y \leftarrow 0$

Αρχή_επανάληψης

$x \leftarrow x + 1$

$y \leftarrow y + x$

Μέχρις_ότου $y > 15$

Εμφάνισε y

β) Να μετατραπεί σε ισοδύναμο διάγραμμα ροής κάνοντας αποκλειστική χρήση των δομών επανάληψης ΟΣΟ αντί της ΜΕΧΡΙΣ_ΟΤΟΥ.

5. α) Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής

$y \leftarrow 8$

$S \leftarrow 0$

Όσο $y < 12$ επανάλαβε

Διάβασε x

$S \leftarrow S + x + y$

$y \leftarrow y + 1$

Τέλος_επανάληψης

Εμφάνισε S

β) Να μετατραπεί σε ισοδύναμο διάγραμμα ροής κάνοντας αποκλειστική χρήση των δομών επανάληψης 1) ΜΕΧΡΙΣ_ΟΤΟΥ αντί της ΟΣΟ και 2) ΓΙΑ αντί της ΟΣΟ

6. α) Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής

$\pi \leftarrow 0$

Για i από 1 μέχρι 14 με_βήμα 3

$y \leftarrow 0$

Όσο $y < 8$ επανάλαβε

Διάβασε x

Αν $x < 0$ τότε

$\pi \leftarrow \pi + 1$

```

y ← y + 1
Τέλος_αν
y ← y + 1
Τέλος_επανάληψης
Τέλος_επανάληψης
Εμφάνισε π, γ

```

β) Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής κάνοντας αποκλειστική χρήση των δομών επανάληψης 1) ΜΕΧΡΙΣ_ΌΤΟΥ αντί της ΟΣΟ και 2) ΓΙΑ αντί της ΌΣΟ.

7. α) Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου σε ισοδύναμο διάγραμμα ροής.

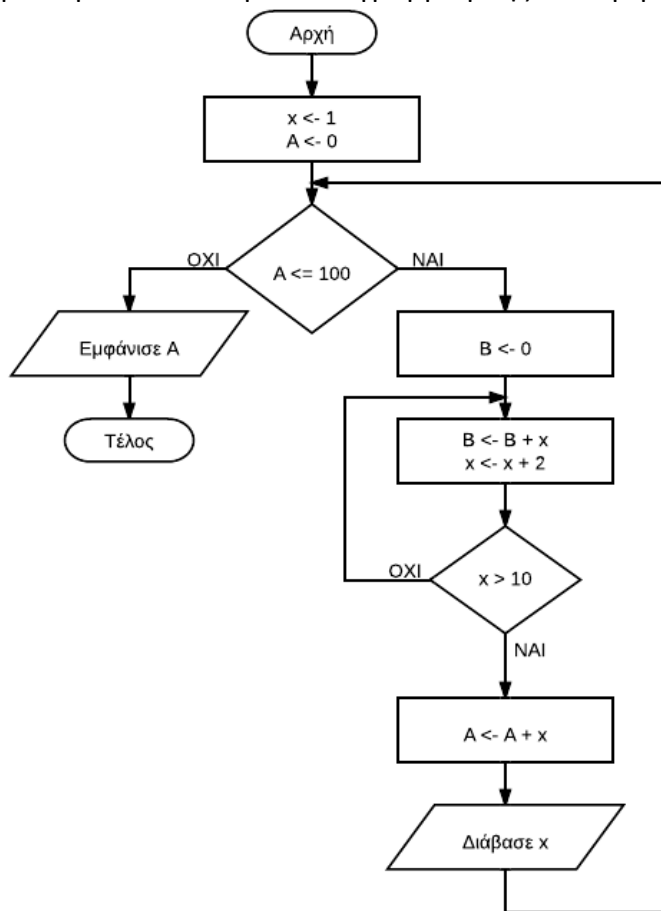
```

Σ ← 0
Αρχή_επανάληψης
Διάβασε x
Όσο x > 0 επανάλαβε
    Εμφάνισε x
    x ← x - 3
    Σ ← Σ + x
Τέλος_επανάληψης
Μέχρις_ότου Σ > 200

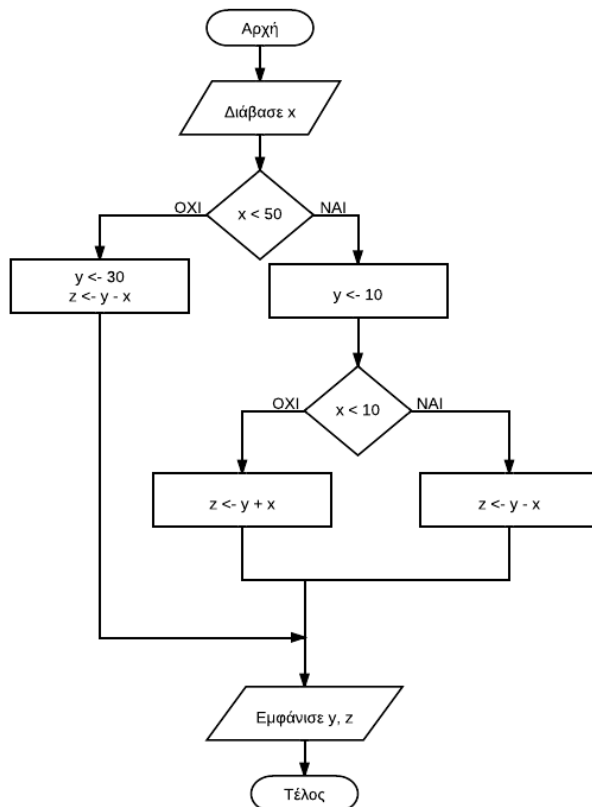
```

β) Να μετατραπεί ο ακόλουθος αλγόριθμος σε ισοδύναμο διάγραμμα ροής κάνοντας αποκλειστική χρήση των δομών επανάληψης
 1) ΟΣΟ αντί της ΜΕΧΡΙΣ_ΌΤΟΥ και
 2) ΜΕΧΡΙΣ_ΌΤΟΥ αντί της ΌΣΟ.

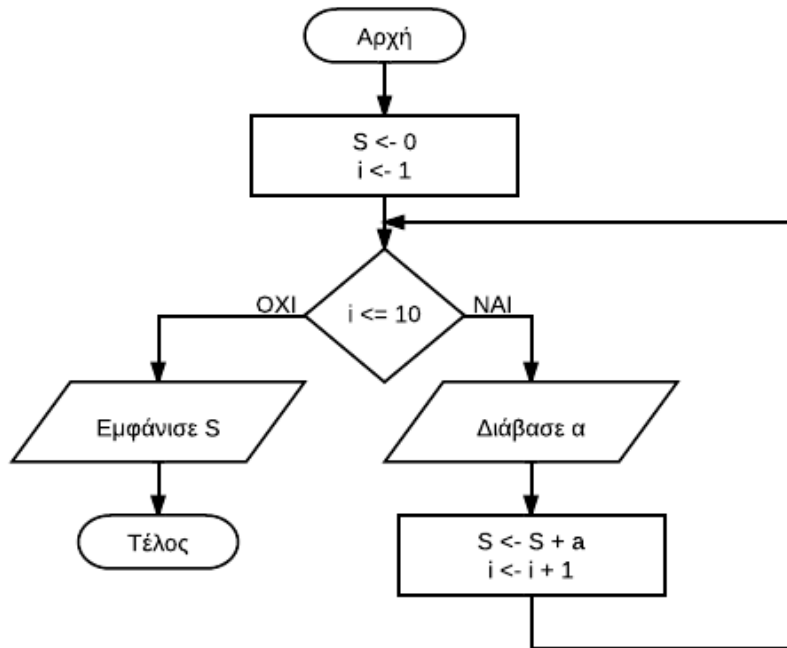
8. Να μετατραπεί το επόμενο διάγραμμα ροής σε αλγόριθμο



9. Να μετατραπεί το επόμενο διάγραμμα ροής σε αλγόριθμο.



10. Να μετατραπεί το επόμενο διάγραμμα ροής σε αλγόριθμο.



11. Να σχεδιάσετε λογικό διάγραμμα, το οποίο να δέχεται από το χρήστη ακέραιους αριθμούς για ένα πίνακα 10 γραμμών και 10 στηλών. Να υπολογίζει και να παρουσιάζει το μικρότερο αριθμό του πίνακα. Επίσης να παρουσιάζει τον πίνακα σε μορφή 10x10.

12. Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται ένα πλήθος θετικών ακεραίων αριθμών και να υπολογίζει και παρουσιάζει το άθροισμα τους και τον μέσο όρο τους. Η διαδικασία να τερματίζεται με την εισαγωγή αρνητικού αριθμού.

13. Να μετατραπούν τα τμήματα αλγορίθμου από Όσο σε Για ... από ... μέχρι.

α)

$A \leftarrow 10$

Όσο $A \leq 22$ επανάλαβε

Διάβασε B

Εμφάνισε A, B

$A \leftarrow A + 3$

Τέλος_επανάληψης

Εμφάνισε A

β)

$A \leftarrow 10$

Όσο $A < 22$ επανάλαβε

Διάβασε B

Εμφάνισε A, B

$A \leftarrow A + 3$

Τέλος_επανάληψης

Εμφάνισε A

14. Να μετατραπούν τα τμήματα αλγορίθμου από Όσο... Επανάλαβε σε Για ... από ... μέχρι.

α)

$j \leftarrow 1$

$S \leftarrow 0$

Όσο $j \leq 10$ επανάλαβε

$j \leftarrow j + 1$

$S \leftarrow S + j$

Τέλος_επανάληψης

Εμφάνισε S

β)

$k \leftarrow A \bmod 20$

$S \leftarrow 0$

Όσο $k > 10$ επανάλαβε

$k \leftarrow k - 2$

$S \leftarrow S + k$

Τέλος_επανάληψης

Εμφάνισε S

ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ ΑΛΓΟΡΙΘΜΩΝ ΟΜΑΔΑ Α΄**ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ ΜΕ ΑΛΓΟΡΙΘΜΟ**

1. Σε μια τάξη υπάρχουν 20 μαθητές που έχουν βαθμολογηθεί σε 10 μαθήματα. Να γραφεί αλγόριθμος που διαβάζει το βαθμό κάθε μαθητή, να υπολογίζει το μέσο όρο κάθε μαθητή και να βρίσκει και να εμφανίζει τον καλύτερο βαθμό.
2. Ένας μαθητής εξετάστηκε σε 13 μαθήματα στις πανελλαδικές εξετάσεις. Να γραφεί αλγόριθμος ο οποίος :
 - α) να διαβάζει τη βαθμολογία του μαθητή σε κάθε μάθημα
 - β) να υπολογίζει και να εμφανίζει τη μικρότερη και τη μεγαλύτερη βαθμολογία που έλαβε στις παραπάνω εξετάσεις.
3. Ένας τελειόφοιτος της Γ΄ Λυκείου έδωσε γραπτές εξετάσεις σε 13 μαθήματα. Αν η διαφορά του μέσου όρου του προφορικού βαθμού και του γραπτού βαθμού είναι μεγαλύτερη των δύο μονάδων, τότε γίνεται προσαρμογή του προφορικού βαθμού στις 2 μονάδες πάνω από το γραπτό βαθμό, ενώ αν η διαφορά γραπτού και προφορικού βαθμού είναι μεγαλύτερη των 2 μονάδων, γίνεται προσαρμογή του προφορικού στις 2 μονάδες κάτω από το γραπτό βαθμό. Στον τελικό βαθμό κάθε μαθήματος, ο προφορικός βαθμός συμμετέχει με 30% και ο γραπτός με 70%. Να γραφεί αλγόριθμος ο οποίος :
 - α) να διαβάζει το μέσο όρο του προφορικού βαθμού και τον γραπτό βαθμό για κάθε μάθημα
 - β) να υπολογίζει την ενδεχόμενη προσαρμογή
 - γ) να υπολογίζει και να εμφανίζει τον τελικό βαθμό για κάθε μάθημα
4. Να γραφεί αλγόριθμος που υπολογίζει και να εμφανίζει το άθροισμα των τετραγώνων των περιττών αριθμών από το 1 ως το 100. ($1^2 + 3^2 + 5^2 + 7^2 \dots$).
5. Να γραφεί αλγόριθμος ο οποίος εμφανίζει τις δεκάδες από το 0 ως το 100 (10,20,30 κλπ), και να υπολογίζει το άθροισμα τους.
6. Στην Ολυμπιάδα Πληροφορικής έλαβαν μέρος 200 υποψήφιοι και βαθμολογήθηκαν σε ακέραια κλίμακα από το 1 ως το 500. Στον δεύτερο γύρο προκρίνονται όσοι υποψήφιοι έχουν βαθμολογία πάνω από 300 βαθμούς.
Να γραφεί αλγόριθμος ο οποίος :
 - α) να διαβάζει τη βαθμολογία κάθε υποψηφίου
 - β) να υπολογίζει και να εμφανίζει πόσοι από τους υποψηφίους προκρίθηκαν στο 2^ο γύρο
 - γ) να υπολογίζει και να εμφανίζει τη βαθμολογία του καλύτερου και του χειρότερου υποψηφίου από όσους προκρίθηκαν.
7. Να γραφεί αλγόριθμος ο οποίος υπολογίζει και εμφανίζει τους αριθμούς από το 100 ως το 999, των οποίων το άθροισμα των ψηφίων τους είναι μεγαλύτερο ή ίσο του 20.
8. Να γραφεί αλγόριθμος ο οποίος θα εκτελεί κάποια από τις βασικές πράξεις πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση ανάμεσα σε δύο ακραίους και θα εμφανίζει το αποτέλεσμα στην οθόνη.

Ο αλγόριθμος θα ελέγχεται από το παρακάτω μενού επιλογής και θα σταματά όταν ο χρήστης επιλέξει Έξοδο. (Να γίνεται έλεγχος εγκυρότητας δεδομένων για την επιλογή του χρήστη) :

1. Πρόσθεση
2. Αφαίρεση
3. Πολλαπλασιασμός
4. Διαίρεση
5. Έξοδος

Δώσε επιλογή :

9. Να γραφεί αλγόριθμος ο οποίος δέχεται ένα άγνωστο πλήθος ακέραιων περιττών αριθμών, να εμφανίζει το μέσο όρο των περιττών αριθμών που δόθηκαν με την εξής μορφή : π.χ. «Ο Μ.Ο. των 15 αριθμών είναι 45». Η διαδικασία να σταματά όταν δοθεί ένας άρτιος αριθμός ή το μηδέν. Αν δοθεί μόνο άρτιος ή το μηδέν να εμφανίζεται το μήνυμα «Δε δόθηκαν περιττοί αριθμοί».
10. Κάποιος κατέθεσε 3.000 ευρώ σε μια τράπεζα με ετήσιο επιτόκιο 5%. Να γράψετε αλγόριθμο ο οποίος:
 - α) υπολογίζει και εμφανίζει σε πόσα χρόνια το κεφάλαιό του θα ξεπεράσει τα 4.000 ευρώ.
 - β) υπολογίζει και εμφανίζει πόσο θα είναι το κεφάλαιό του μετά από 10 χρόνια.

11. Η χρέωση στους λογαριασμούς τηλεφωνίας υπολογίζεται από τον παρακάτω πίνακα:

Πάγιο	12 €	
Αστικές Μονάδες	0,02 €/μονάδα	
Υπεραστικές Μονάδες	0 - 200	0,14 €/μονάδα
	201 - 400	0,10 €/μονάδα
	401 - ΥΜ	0,09 €/μονάδα
ΦΠΑ (επί του συνόλου): 24%		

Να γραφεί αλγόριθμος που θα διαβάσει τις αστικές (ΑΜ) και τις υπεραστικές μονάδες (ΥΜ) και θα εμφανίζει τη χρέωση (Χ) του συνδρομητή.

12. Μια εταιρεία αποφασίζει να τροποποιήσει τις μηνιαίες αποδοχές των υπαλλήλων της. Έτσι, όσοι παίρνουν κάθε μήνα πάνω από 3000€ θα έχουν μείωση 20%, ενώ όσοι αμείβονται κάτω από 1000€ θα έχουν αύξηση 10%.
Να δημιουργήσετε αλγόριθμο που:
 1. Θα δέχεται όνομα υπαλλήλου και μηνιαία απολαβή.
 2. Θα υπολογίζει τα χρήματα που έδινε η εταιρεία στους υπαλλήλους της πριν και μετά τις αλλαγές κάθε μήνα, εμφανίζοντας παράλληλα μήνυμα με το εάν συμφέρει την εταιρεία η αλλαγή αυτή.
 3. Θα τερματίζεται η διαδικασία εισαγωγής δεδομένων, όταν δοθεί ως όνομα υπαλλήλου η λέξη «ΤΕΛΟΣ».
 4. Θα εμφανίζει τα ονόματα των καλύτερα και χειρότερα αμειβόμενων υπαλλήλων της εταιρείας.
 5. Τέλος, θα εμφανίζει τον Μέσο Μισθό μετά τις αλλαγές στη μισθοδοσία, καθώς και το ποσοστό – επί του συνόλου των υπαλλήλων – όσων η αμοιβή και μετά την αλλαγή δεν υπερβαίνει τα 1000€.

**ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ ΑΛΓΟΡΙΘΜΩΝ ΟΜΑΔΑ Β'
ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ ΜΕ ΑΛΓΟΡΙΘΜΟ**

1. Σε ένα νοσοκομείο οι ασθενείς που νοσηλεύονται έχουν 3 ασφάλειες: «ΤΕΒΕ», «ΙΚΑ», «ΔΗΜΟΣΙΟ». Να γραφεί αλγόριθμος που θα διαβάζει για κάθε ασθενή που εξέρχεται από το νοσοκομείο το είδος ασφάλειας (ΕΑ) και τον αριθμό των ημερών (Η) που διέμεινε στο νοσοκομείο και θα:
α) κάνει έλεγχο εγκυρότητας ώστε οι ημέρες να είναι θετικός αριθμός,
β) υπολογίζει και εμφανίζει πόσοι ασθενείς υπήρχαν σε κάθε κατηγορία,
γ) υπολογίζει και εμφανίζει το σύνολο των ημερών που έμειναν στο νοσοκομείο όλοι οι ασθενείς του ΔΗΜΟΣΙΟΥ.
δ) Γνωρίζοντας ότι το ΤΕΒΕ δίνει 25 Ευρώ αποζημίωση για κάθε ημέρα διαμονής, το ΙΚΑ 45 Ευρώ και το ΔΗΜΟΣΙΟ 30 Ευρώ να εμφανιστεί για κάθε ασθενή πόσα χρήματα θα πάρει για τις ημέρες που έμεινε στο νοσοκομείο.
Ο αλγόριθμος θα τερματίζει όταν δοθεί ως είσοδος μη αποδεκτή ασφάλεια.
2. Η ΔΕΗ εφαρμόζει την εξής μηνιαία τιμολογιακή πολιτική:
 - 1) Πάγιο 25 Ευρώ
 - 2) Για κάθε κιλοβατώρα χρεώνει 1 Ευρώ και 28 λεπτά αν έχει γίνει συνολική κατανάλωση μέχρι και 200 κιλοβατώρες. Αν έχουν καταναλωθεί περισσότερες από 200 κιλοβατώρες, οι πρώτες ακολουθούν την προηγούμενη χρέωση, ενώ καθεμία από τις υπόλοιπες χρεώνεται προς 1.10 Ευρώ.
 - 3) Αν κάποιος καταναλωτής είχε βλάβες, τότε πληρώνει έξτρα 5.5 Ευρώ.Να γραφεί αλγόριθμος που για καθέναν από τους δημότες μιας πόλης θα διαβάζει το ονοματεπώνυμό του, τη μηνιαία κατανάλωσή του σε κιλοβατώρες και μία από τις λέξεις "ΝΑΙ" ή "ΟΧΙ". Αν διαβαστεί "ΟΧΙ" τότε ο δημότης δεν έχει βλάβες.
Ο αλγόριθμος να υπολογίζει και να εμφανίζει:
 - α) τη χρέωση κάθε δημότη μετά από το ονοματεπώνυμό του,
 - β) τα συνολικά έσοδα της ΔΕΗ,
 - γ) το δημοτικό φόρο που είναι το $1/500$ των συνολικών εσόδων της ΔΕΗ.Η είσοδος δεδομένων να επαναλαμβάνεται όσο δίνονται ως δεδομένα θετικοί αριθμοί και μία από τις δύο επιτρεπτές λέξεις "ΝΑΙ" ή "ΟΧΙ".
3. Ένας μαθητής θέλει να αγοράσει κάποια CD (ψηφιακός δίσκος) για να εμπλουτίσει τη συλλογή του. Διαθέτει 83€ που έχει συγκεντρώσει από τα γενέθλιά του. Έτσι πηγαίνει σε ένα κατάστημα και αγοράζει CD μέχρι να μην του περισσεύουν άλλα χρήματα για την αγορά ψηφιακών δίσκων.
Κατά την αγορά κάθε δίσκου εισάγει στον αλγόριθμο που εκτελείται στο κινητό του το κόστος του δίσκου και ο αλγόριθμος εμφανίζει το μήνυμα "Τέλος", όταν δεν μπορεί να αγοραστεί άλλος δίσκος. Αν κατά τη διάρκεια των αγορών, φθάσει στην αγορά του 10ου δίσκου, τότε γίνεται 40% έκπτωση στο κόστος κάθε δίσκου από τον 10ο και έπειτα.
 - α. Να εμφανιστεί ο συνολικός αριθμός δίσκων και το συνολικό κόστος τους.
 - β. Να εμφανίζει τυχόν περίσσειμα χρημάτων ή το μήνυμα «Δεν υπάρχουν ρέστα», ανάλογα με την περίπτωση.

4. Για να γίνει μια επιστημονική εργασία αποδεκτή για δημοσίευση σε κάποιο επιστημονικό σύγγραμμα, αξιολογείται από δύο κριτές και το αποτέλεσμα είναι "Εγκρίνεται" ή "Απορρίπτεται". Ένα σύνολο εργασιών στάλθηκε στον υπεύθυνο ενός νέου συγγράμματος ο οποίος αντιστοιχεί κάποιον αλφαριθμητικό κωδικό σε κάθε εργασία και την αποστέλλει σε δύο κριτές.

Στο τέλος της διαδικασίας αξιολόγησης, για να αποφασιστούν οι εργασίες που θα συμπεριληφθούν στο σύγγραμμα, εισάγεται σε αλγόριθμο ο αλφαριθμητικός κωδικός κάθε εργασίας, ο αριθμός σελίδων της και το αποτέλεσμα καθενός από τους δύο κριτές. Η διαδικασία ολοκληρώνεται όταν δοθεί ως κωδικός ο χαρακτήρας 0 (μηδέν).

- α) Ο αλγόριθμος ελέγχει για κάθε εργασία το αποτέλεσμα των κριτών και αν συμφωνούν μεταξύ τους, τότε εμφανίζει το αποτέλεσμα και τον κωδικό της εργασίας. Αν δεν συμφωνούν τα δύο δεδομένα, τότε διαβάζεται το αποτέλεσμα από την αξιολόγηση ενός τρίτου κριτή και η εργασία εγκρίνεται ή απορρίπτεται ανάλογα με το ποιο αποτέλεσμα υπερισχύει ανάμεσα στις τρεις αξιολογήσεις. Και σε αυτή την περίπτωση εμφανίζει το αποτέλεσμα και τον κωδικό της εργασίας.
- β) Εμφανίζει τον αριθμό των εργασιών που τελικά εγκρίθηκαν και τον αριθμό των εργασιών που απορρίφθηκαν.
- γ) Εμφανίζει το σύνολο των σελίδων του συγγράμματος με βάση το σύνολο των σελίδων των εγκεκριμένων εργασιών.
5. Μια εταιρεία αποφασίζει να τροποποιήσει τις μηνιαίες αποδοχές των υπαλλήλων της. Έτσι, όσοι παίρνουν κάθε μήνα πάνω από 3000€ θα έχουν μείωση 20%, ενώ όσοι αμείβονται κάτω από 1000€ θα έχουν αύξηση 10%.
- Να δημιουργήσετε αλγόριθμο που:
1. Θα δέχεται όνομα υπαλλήλου και μηνιαία απολαβή.
 2. Θα υπολογίζει τα χρήματα που έδινε η εταιρεία στους υπαλλήλους της πριν και μετά τις αλλαγές κάθε μήνα, εμφανίζοντας παράλληλα μήνυμα με το εάν συμφέρει την εταιρεία η αλλαγή αυτή.
 3. Θα τερματίζεται η διαδικασία εισαγωγής δεδομένων, όταν δοθεί ως όνομα υπαλλήλου η λέξη «ΤΕΛΟΣ».
 4. Θα εμφανίζει τα ονόματα των καλύτερα και χειρότερα αμειβόμενων υπαλλήλων της εταιρείας.
 5. Τέλος, θα εμφανίζει τον Μέσο Μισθό μετά τις αλλαγές στη μισθοδοσία, καθώς και το ποσοστό των υπαλλήλων όσων η αμοιβή και μετά την αλλαγή δεν υπερβαίνει τα 1000 €.

ΑΛΥΤΕΣ ΟΜΑΔΕΣ ΑΣΚΗΣΕΩΝ ΣΕ ΑΛΓΟΡΙΘΜΟ

- 1 Ένα εργοστάσιο έχει 200 υπαλλήλους. Για κάθε ένα από τους υπαλλήλους εισάγονται από το πληκτρολόγιο ο μισθός, η ηλικία του καθώς επίσης και το όνομα του. Να γραφεί αλγόριθμος ο οποίος:
- Α. Θα διαβάζει το όνομα την ηλικία και το μισθό κάθε υπαλλήλου (όχι έλεγχο).
- Β. Θα εμφανίζει πόσα άτομα που έχουν το όνομα Μανώλης πληρώνονται με μισθό άνω των 1000 ευρώ και πόσο είναι ο μέσος μισθός των υπαλλήλων αυτών.
- Γ. Αν υποθέσουμε ότι οι υπάλληλοι αυτοί συνταξιοδοτούνται στην ηλικία των 65 ετών, να εμφανίζεται το όνομα κάθε υπαλλήλου που πρόκειται να συνταξιοδοτηθεί μέσα στην επόμενη επταετία (7 έτη).

2. Μια αλυσίδα σούπερ μάρκετ έχει εγκαταστάσεις σε 35 πόλεις σε όλη την Ελλάδα. Οι μηνιαίοι τζίροι (συνολικές πωλήσεις) για κάθε κατάστημα για διάστημα ενός χρόνου (12 μήνες) καταγράφεται σε ένα πίνακα Α. Τα ονόματα των πόλεων στις οποίες έχει καταστήματα η αλυσίδα βρίσκονται σε ένα πίνακα Β. Σε ένα πίνακα Γ βρίσκονται τα ονόματα των διευθυντών του κάθε καταστήματος. Να γραφεί αλγόριθμος ο οποίος:
1. Γεμίζει τους πίνακες Α, Β και Γ με τιμές που δίνει ο χρήστης από το πληκτρολόγιο.
 2. Σε ποια πόλη και σε ποιο μήνα σημειώθηκαν οι μεγαλύτερες πωλήσεις του έτους για την εταιρία; (όχι ισοβαθμία)
 3. Αν πρέπει να πάρει bonus παραγωγικότητας, 2% επί του συνολικού τζίρου του, ο διευθυντής με τις μεγαλύτερες πωλήσεις συνολικά, ποιος θα είναι αυτός ο διευθυντής (το ονομά του) και τι ποσό πρόκειται να πάρει σαν bonus; Να σημειωθεί ότι δεν υπάρχουν ισοβαθμίες στους τζίρους των καταστημάτων.
 4. Να διαβάζεται από το πληκτρολόγιο το όνομα ενός διευθυντή και να εμφανίζει αν ο διευθυντής αυτός υπάρχει στο δυναμικό της επιχείρησης ή όχι. Αν ναι τότε σε ποιο κατάστημα είναι διευθυντής;
4. Δίνονται ο πληθωρισμός, τα έσοδα εξαγωγών και το κατά κεφαλήν εισόδημα για καθεμιά από τις 15 χώρες της Ευρωπαϊκής Ένωσης. Να αναπτύξετε αλγόριθμο που
- α. Θα διαβάζει τα παραπάνω δεδομένα.
 - β. Θα βρίσκει το μέσο όρο πληθωρισμού.
 - γ. Θα εμφανίζει τη χώρα με το μικρότερο κατά κεφαλήν εισόδημα.
 - δ. Θα ταξινομεί τους πίνακες σε φθίνουσα σειρά ως προς τα έσοδα εξαγωγών.
 - ε. Θα εμφανίζει τα παραπάνω ταξινομημένα στοιχεία.

ΚΕΦΑΛΑΙΟ 6

6.1 Η έννοια του προγραμματισμού

Η επίλυση ενός προβλήματος με τον υπολογιστή περιλαμβάνει, όπως έχει ήδη αναφερθεί, τρία εξίσου σημαντικά στάδια.

- Τον ακριβή προσδιορισμό του προβλήματος.
- Την ανάπτυξη του αντίστοιχου αλγορίθμου.
- Τη διατύπωση του αλγορίθμου σε κατανοητή μορφή από τον υπολογιστή.

Ο **προγραμματισμός** ασχολείται με το τρίτο αυτό στάδιο, τη δημιουργία του προγράμματος δηλαδή του συνόλου των εντολών που πρέπει να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος. Το πρόγραμμα, το οποίο γράφεται σε κάποια γλώσσα προγραμματισμού, δεν είναι απλά η υλοποίηση του αλγορίθμου, αλλά βασικό στοιχείο του είναι τα δεδομένα και οι δομές δεδομένων επί των οποίων ενεργεί. Αναφέρθηκε ήδη ότι οι αλγόριθμοι και οι δομές δεδομένων είναι μια αδιάσπαστη ενότητα.

Συμπερασματικά ο **προγραμματισμός** είναι η διατύπωση του αλγορίθμου σε μορφή κατανοητή από τον Η/Υ ώστε να τον εκτελέσει («τρέξει» όπως λέμε στην ορολογία της Πληροφορικής).

Η διατύπωση γίνεται χρησιμοποιώντας μία γλώσσα προγραμματισμού.

Ο προγραμματισμός είναι αυτός που δίνει την εντύπωση ότι οι υπολογιστές είναι έξυπνες μηχανές που επιλύουν τα πολύπλοκα προβλήματα.

Η εντύπωση αυτή όμως είναι απλώς μία ψευδαίσθηση. Ο υπολογιστής, ως γνωστόν, είναι μία μηχανή που καταλαβαίνει μόνο δύο καταστάσεις, οι οποίες αντιπροσωπεύονται με δύο αριθμούς, το 0 και το 1, τα ψηφία του δυαδικού συστήματος.

Το μόνο πράγμα που κάνει ο υπολογιστής είναι στοιχειώδεις ενέργειες σε ακολουθίες αυτών των δύο ψηφίων, αλλά αυτές τις ενέργειες τις εκτελεί με ασύλληπτη ταχύτητα.

Ο υπολογιστής μπορεί απλά να αποθηκεύει στη μνήμη τις ακολουθίες των δυαδικών ψηφίων, να τις ανακτά, να κάνει στοιχειώδεις αριθμητικές πράξεις με αυτές και να τις συγκρίνει.

Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία του ανθρώπου (προγραμματιστή) με τη μηχανή (υπολογιστή).

6.2 Ιστορική αναδρομή

Κατηγορίες γλωσσών προγραμματισμού

6.2.1 Γλώσσες μηχανής	Ένα πρόγραμμα σε γλώσσα μηχανής είναι μια ακολουθία δυαδικών ψηφίων, που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Το πρόγραμμα περιέχει εντολές που είναι σε δυαδική μορφή που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Είναι άμεσα κατανοητή από τον Η/Υ (όχι όμως από τον άνθρωπο). Δηλαδή, το πρόγραμμα αποτελείται από ακολουθίες 0 και 1 π.χ. <pre>10101000 00001010 11000000 00000001</pre> Πλεονεκτήματα: • Ταχύτατη εκτέλεση των εντολών. • Δεν απαιτείται μεταφραστικό πρόγραμμα. Μειονεκτήματα: • Το γράψιμο του προγράμματος είναι μία ιδιαίτερα επίπονη και χρονοβόρα διαδικασία. • Απαιτείται βαθιά γνώση της αρχιτεκτονικής του Η/Υ. • Το πρόγραμμα «τρέχει» μόνο στο συγκεκριμένο τύπο του Η/Υ.
-------------------------------------	--

6.2.2 Γλώσσες χαμηλού επιπέδου ή Συμβολικές γλώσσες	Οι εντολές σε συμβολική γλώσσα αποτελούνται από συμβολικά ονόματα που αντιστοιχούν σε εντολές της γλώσσας μηχανής. Οι συμβολικές γλώσσες είναι συνδεδεμένες με την αρχιτεκτονική κάθε υπολογιστή. Οι εντολές που είναι σε μορφή 0 και 1 αντικαθίστανται από μνημονικά (συμβολικά) ονόματα. Για παράδειγμα, η εντολή 100001100 αντικαθίστανται από το ADD. Ένα δείγμα χρήσης θα ήταν: <pre> 10101000 00000010 SUM=\$02 10001100 00000001 INDEX=\$01 00111100 LDA #10 01010001 00000001 STA INDEX 01000011 00000001 CLA 11000000 11111010 LOOP ADD INDEX 10001100 00000010 DEC INDEX </pre>
---	---

	11111111 BNE LOOP STA SUM BRK Ένα πρόγραμμα σε γλώσσα μηχανής, σε συμβολική γλώσσα και σε γλώσσα υψηλού επιπέδου για τον υπολογισμό του αθροίσματος των αριθμών 1 έως 10. Πλεονεκτήματα: • Ταχύτατη εκτέλεση των εντολών. • Η μορφή του προγράμματος είναι καλύτερα κατανοητή από τον άνθρωπο σε σχέση με τη γλώσσα μηχανής. Μειονεκτήματα: • Η αντιστοιχία 1 προς 1 με τις εντολές της γλώσσας παρέμενε. • Απαιτείται η χρήση ενός μεταφραστικού προγράμματος ώστε οι συμβολικές εντολές να μετατραπούν στις αντίστοιχες δυαδικές. Το ειδικό αυτό πρόγραμμα ονομάζεται συμβολομεταφραστής (assembler). • Το γράψιμο του προγράμματος εξακολουθεί να είναι μία ιδιαίτερα επίπονη και χρονοβόρα διαδικασία. • Απαιτείται βαθιά γνώση της αρχιτεκτονικής του Η/Υ. • Το πρόγραμμα «τρέχει» μόνο στο συγκεκριμένο τύπο του Η/Υ.
6.2.3 Γλώσσες υψηλού επιπέδου	Λέγονται έτσι διότι τα προγράμματα διατυπωμένα σε μία τέτοια γλώσσα είναι άμεσα κατανοητά από τον άνθρωπο (αλλά όχι από τον Η/Υ) αφού χρησιμοποιείται μία γλώσσα που είναι αρκετά περιγραφική όπως μία φυσική γλώσσα. Οι γλώσσες υψηλού επιπέδου χρησιμοποιούν ως εντολές απλές λέξεις της αγγλικής γλώσσας ακολουθώντας αυστηρούς κανόνες σύνταξης, οι οποίες μεταφράζονται από τον ίδιο τον υπολογιστή σε εντολές σε γλώσσα μηχανής. Παράδειγμα, <pre>INPUT "Δώσε την τελική τιμή" ; N SUM = 0 ForINDEX = 1 to N SUM = SUM + INDEX Next</pre> Πλεονεκτήματα: Η μορφή του προγράμματος είναι εύκολα κατανοητή από τον άνθρωπο σε σχέση με τη γλώσσα μηχανής ή τη συμβολική γλώσσα. • Το γράψιμο του προγράμματος δεν είναι πλέον μία ιδιαίτερα

	επίπονη και χρονοβόρα διαδικασία όπως συμβαίνει με τη γλώσσα μηχανής ή τη συμβολική γλώσσα. • Δεν απαιτείται σχεδόν καμία γνώση της αρχιτεκτονικής του Η/Υ. Συνεπώς, είναι ανεξάρτητα από την αρχιτεκτονική του Η/Υ. • Το πρόγραμμα «τρέχει» σε όλους τους τύπους Η/Υ αρκεί να υπάρχει το κατάλληλο μεταφραστικό πρόγραμμα. Συνεπώς ένα χαρακτηριστικό τους είναι η μεταφερσιμότητα, δηλαδή ένα πρόγραμμα υψηλού επιπέδου να εκτελείται, με ελάχιστες μετατροπές, σε πολλούς τύπους Η/Υ. • Η εκμάθηση της γλώσσας είναι εύκολη. • Η διόρθωση λαθών και η συντήρηση των προγραμμάτων είναι ευκολότερη. Μειονεκτήματα: • Απαιτείται η χρήση ενός μεταφραστικού προγράμματος ώστε οι εντολές να μετατραπούν σε πολλές δυαδικές εντολές (δεν έχουμε εδώ αντιστοιχία 1 προς 1). Έχουμε δύο ειδών μεταφραστικά προγράμματα: τους μεταγλωττιστές (compilers) και τους διερμηνείς (interpreters). • Το πρόγραμμα «τρέχει» πιο αργά σε σχέση με τα προγράμματα των συμβολικών γλωσσών ή της γλώσσας μηχανής.
--	---

Κατηγοριοποίηση γλωσσών υψηλού επιπέδου:

- **Διαδικασιακές ή αλγοριθμικές γλώσσες (Procedural):** λέγονται έτσι διότι επιτρέπουν την εύκολη υλοποίηση αλγορίθμων π.χ. Pascal, Basic.
- **Αντικειμενοστραφείς γλώσσες (object – oriented)** π.χ. C++
- **Συναρτησιακές γλώσσες (functional)** π.χ. LISP
- **Μη-διαδικασιακές γλώσσες** π.χ. PROLOG
- **Γλώσσες ερωταπαντήσεων (Query languages) ή 4ης γενιάς** π.χ. SQL

Μία άλλη κατηγοριοποίηση των γλωσσών υψηλού επιπέδου είναι η εξής:

- **Γλώσσες γενικής χρήσης:** Σκοπός τους είναι να επιλύουν πάσης φύσεως προβλήματα (αριθμητικά, εμπορικά, επιστημονικά). Τέτοιες είναι η Basic, Pascal. Μερικές γλώσσες, όμως, έχουν δημιουργηθεί αποκλειστικά για να επιλύουν ευκολότερα συγκεκριμένους τύπους προβλημάτων όπως:
 - **Γλώσσες επιστημονικής κατεύθυνσης:** π.χ. FORTRAN
 - **Γλώσσες εμπορικής κατεύθυνσης:** π.χ. COBOL.
- **Γλώσσες προγραμματισμού συστημάτων** π.χ. C
- **Γλώσσες τεχνητής νοημοσύνης** π.χ. LISP, PROLOG.
- **Γλώσσες ειδικής χρήσης.** Σκοπός τους είναι να επιλύουν ειδικού τύπου προβλήματα όπως διαχείριση Βάσεων Δεδομένων κ.α. π.χ. SQL.

Διαδικασιακές γλώσσες κατά χρονολογική σειρά εμφάνισης	
FORTRAN (1957)	Κατάλληλη για επίλυση επιστημονικών προβλημάτων (αριθμητικές εφαρμογές).
COBOL (1960)	Κατάλληλη για επίλυση εμπορικών προβλημάτων (εφαρμογές μισθοδοσίας, λογιστικές κλπ).
ALGOL (1960)	Δημιουργήθηκε με σκοπό την ανάπτυξη προγραμμάτων κυρίως για επιστημονικές εφαρμογές ως ανταγωνιστική της Fortran. Δεν χρησιμοποιήθηκε, όμως, ευρέως στην πράξη. Πρωτοπαρουσίασε τις αρχές του δομημένου προγραμματισμού.
PL/1 (μέσα '60)	Προσπάθησε να συνδυάσει τις δυνατότητες των γλωσσών προσανατολισμένων για εμπορικές και επιστημονικές εφαρμογές χωρίς όμως να γνωρίσει επιτυχία.
BASIC (μέσα '60)	Δημιουργήθηκε με σκοπό την εκπαίδευση των αρχάριων στον προγραμματισμό. Σκοπός της BASIC είναι να γράφονται μικρά προγράμματα που κατόπιν εκτελούνται με τη βοήθεια διερμηνέα (interpreter). Στις μέρες μας αποτελεί μία πανίσχυρη, γενικής χρήσης γλώσσα.
PASCAL (1970)	Γλώσσα γενικής χρήσης. Στηρίχθηκε στην ALGOL. Είναι η καταλληλότερη γλώσσα για να μάθει κάποιος δομημένο προγραμματισμό.
C (αρχές '70)	Περιέχει αρκετά κοινά χαρακτηριστικά με τη Pascal για την ανάπτυξη δομημένων εφαρμογών αλλά παράλληλα ενσωματώνει και χαρακτηριστικά γλώσσας χαμηλού επιπέδου. Θεωρείται κατάλληλη για την ανάπτυξη λειτουργικών συστημάτων (π.χ. Unix)
Ada (1979)	Γλώσσα γενικής χρήσης, δίνει έμφαση στο θέμα της αξιοπιστίας των προγραμμάτων. Χρησιμοποιήθηκε πρωτίστως για στρατιωτικές εφαρμογές.

Αντικειμενοστραφείς γλώσσες κατά χρονολογική σειρά εμφάνισης	
Smalltalk (αρχές '80)	Η πρώτη αντικειμενοστραφής γλώσσα με ολοκληρωμένο μάλιστα περιβάλλον ανάπτυξης προγραμμάτων.
C++ (τέλη '80)	Αποτελεί μία μετεξέλιξη της C στο χώρο του αντικειμενοστραφούς προγραμματισμού και χρησιμοποιείται αρκετά στην ανάπτυξη λειτουργικών συστημάτων (π.χ. Windows) αλλά και άλλου τύπου εφαρμογών. Θεωρείται σήμερα μία κορυφαία γλώσσα.
JAVA (μέσα '90)	Γλώσσα ειδικά σχεδιασμένη για την ανάπτυξη εφαρμογών στο Internet. Σκοπός της είναι να γράφονται προγράμματα που θα εκτελούνται, χωρίς μετατροπές, σε Η/Υ με διαφορετικά λειτουργικά συστήματα. Περιέχει αρκετά στοιχεία από τη C++.

C# (2002)	Δημιουργήθηκε ως ανταγωνιστική της JAVA. Σκοπός της είναι να συνδυάσει την ευχρηστία της Basic και τη δυναμική της C++ για την ανάπτυξη εφαρμογών που θα εκτελούνται σε Η/Υ με διαφορετικά λειτουργικά συστήματα.
----------------------------	---

Συναρτησιακές γλώσσες κατά χρονολογική σειρά εμφάνισης	
LISP (μέσα '60)	Δημιουργήθηκε για την ανάπτυξη προγραμμάτων στο χώρο της τεχνητής νοημοσύνης.

Μη Διαδικασιακές γλώσσες κατά χρονολογική σειρά εμφάνισης	
PROLOG (αρχές '70)	Δημιουργήθηκε για την ανάπτυξη προγραμμάτων στο χώρο της τεχνητής νοημοσύνης.

6.2.4 Γλώσσες 4ης γενιάς ή ερωταπαντήσεων	
SQL	Δεν απευθύνεται μόνο σε προγραμματιστές αλλά και χρήστες. Ο χρήστης μπορεί, σχετικά εύκολα, να υποβάλει ερωτήσεις στο σύστημα ή να αναζητά πληροφορίες από μία Βάση Δεδομένων. Παράδειγμα, <pre>SELECT Επώνυμο, Όνομα FROM Μαθητές WHERE Τάξη = "Γ2"</pre> Η παραπάνω πρόταση της γλώσσας SQL θα κάνει μία αναζήτηση στη βάση δεδομένων των μαθητών και θα εμφανίσει τα ονοματεπώνυμα των μαθητών του Γ2 μόνο.

Από τι εξαρτάται η επιλογή μίας γλώσσας προγραμματισμού;

- Από το είδος του προβλήματος (εμπορικό, αριθμητικό κ.α.)
- Από τον Η/Υ στον οποίο θα εκτελεστεί το πρόγραμμα.
- Από τη διαθέσιμη γλώσσα ή προγραμματιστικό περιβάλλον στο οποίο θα αναπτυχθεί το πρόγραμμα.
- Από τις γνώσεις και την εμπειρία του προγραμματιστή.

Με την ευρεία διάδοση των γραφικών περιβαλλόντων επικοινωνίας (π.χ. Windows, MacOS κλπ) δημιουργήθηκαν παραλλαγές κάποιων γλωσσών που απευθύνονται σε αυτά. Τέτοιες είναι η VisualBasic, VisualC++, Delphi (VisualPascal), C# κ.α. Αυτές οι γλώσσες ακολουθούν τη φιλοσοφία του οπτικού και του καθοδηγούμενου από γεγονότα προγραμματισμού χωρίς να απορρίπτουν τις αρχές του δομημένου προγραμματισμού.

Οπτικός προγραμματισμός

Είναι η δυνατότητα να δημιουργούμε, με γραφικό τρόπο, ολόκληρο το περιβάλλον της εφαρμογής όπως για παράδειγμα τα μενού και τα πλαίσια διαλόγου και άλλα παράθυρα της εφαρμογής.

Ο οπτικός προγραμματισμός εκμεταλλεύεται τις δυνατότητες των γραφικών περιβαλλόντων επικοινωνίας (π.χ. Windows, MacOS κλπ).

Καθοδηγούμενος από γεγονότα προγραμματισμός

Είναι η δυνατότητα να εκτελούνται οι διάφορες λειτουργίες του προγράμματος με την ενεργοποίηση ενός γεγονότος. Για παράδειγμα, αν κάνουμε κλικ σε κάποια εντολή ενός μενού ή σε κάποιο κουμπί σε ένα παράθυρο της εφαρμογής τότε θα εκτελεστεί μία λειτουργία..

Τα σύγχρονα προγραμματιστικά περιβάλλοντα είναι κτισμένα πάνω στις αρχές του οπτικού και καθοδηγούμενου από γεγονότα προγραμματισμού.

Ποιά είναι η καλύτερη γλώσσα προγραμματισμού;

Στην ιστορία του προγραμματισμού έχουν αναπτυχθεί χιλιάδες γλώσσες και αυτή τη στιγμή χρησιμοποιούνται μερικές εκατοντάδες. Υπάρχουν γλώσσες κατάλληλες για ανάπτυξη ειδικών εφαρμογών και άλλες κατάλληλες για γενική χρήση. Υπάρχουν γλώσσες κατάλληλες για εκπαίδευση και άλλες για ανάπτυξη εμπορικών εφαρμογών. Γλώσσες που επιτρέπουν την εύκολη ανάπτυξη εφαρμογών σε γραφικό περιβάλλον και άλλες που εκμεταλλεύονται τα παράλληλα συστήματα. Υπάρχουν γλώσσες πολύ ισχυρές αλλά πολύπλοκες και γλώσσες χωρίς μεγάλες δυνατότητες αλλά απλές και εύκολες στην εκμάθηση. Ο προγραμματιστής καλείται να επιλέξει την "καλύτερη" γλώσσα για να υλοποιήσει το πρόγραμμα.

Μπορούμε να ισχυριστούμε με βεβαιότητα ότι μία γλώσσα προγραμματισμού που να είναι αντικειμενικά καλύτερη από τις άλλες δεν υπάρχει, ούτε πρόκειται να υπάρξει.

Η επιλογή της γλώσσας για την ανάπτυξη μιας εφαρμογής εξαρτάται από το είδος της εφαρμογής, το υπολογιστικό περιβάλλον στο οποίο θα εκτελεστεί, τα προγραμματιστικά περιβάλλοντα που διαθέτουμε και κυρίως τις γνώσεις του προγραμματιστή. Συνήθως ο προγραμματιστής επιλέγει μία γλώσσα, που φυσικά επιτρέπει και διευκολύνει την ανάπτυξη του είδους της εφαρμογής στο συγκεκριμένο περιβάλλον με βάση όμως τις προσωπικές του γνώσεις και προτιμήσεις.

6.3 Φυσικές και τεχνητές γλώσσες

Οι γλώσσες προγραμματισμού είναι τεχνητές γλώσσες που απευθύνονται σε ανθρώπους που επιθυμούν να επικοινωνήσουν με τον Η/Υ.

Κάθε γλώσσα προγραμματισμού προσδιορίζεται από:

- Το αλφάβητό της
- Το λεξιλόγιό της
- Τη γραμματική της
- Τη σημασιολογία της (Semantics)

Αλφάβητο γλώσσας

Ως αλφάβητο ορίζουμε το σύνολο των αποδεκτών χαρακτήρων της γλώσσας. Από τους χαρακτήρες αυτούς σχηματίζονται οι λέξεις της γλώσσας.

Σε μία φυσική γλώσσα, όπως τα Ελληνικά, οι αποδεκτοί χαρακτήρες είναι τα γράμματα Α-Ω (κεφαλαία και πεζά), τα ψηφία 0-9 και τα σημεία στίξης.

Λεξιλόγιο γλώσσας

Το λεξιλόγιο μίας γλώσσας περιλαμβάνει όλες τις έγκυρες και αποδεκτές λέξεις. Στην ουσία, είναι ένα υποσύνολο από όλες τις δυνατές ακολουθίες που μπορούμε να σχηματίσουμε από τα στοιχεία του αλφαβήτου.

Σε μία φυσική γλώσσα, όπως τα Ελληνικά, η λέξη ΔΙΑΒΑΖΩ είναι αποδεκτή ενώ η λέξη ΖΩΒΑΓΩ όχι.

Η γραμματική περιλαμβάνει το **τυπολογικό** και το **συντακτικό**.

Το **τυπολογικό** ορίζει τους κανόνες σύμφωνα με του οποίους μία λέξη θα είναι αποδεκτή.

Για παράδειγμα, στην ελληνική γλώσσα για τη λέξη "δίνω", αποδεκτές μορφές είναι και το "δίνεις", "δίνουν" αλλά όχι το "δίνου".

Το **συντακτικό** είναι ένα σύνολο κανόνων που ορίζει το πώς πρέπει να σχηματίζονται οι προτάσεις από τις λέξεις της γλώσσας ώστε οι προτάσεις αυτές να είναι έγκυρες και αποδεκτές.

Σε μία γλώσσα προγραμματισμού αυτό που ενδιαφέρει είναι η σωστή σύνταξη των εντολών.

Γενικά η γνώση του συντακτικού επιτρέπει τη δημιουργία σωστών προτάσεων στις φυσικές γλώσσες, ενώ στις γλώσσες προγραμματισμού τη δημιουργία σωστών εντολών.

Σημασιολογία της γλώσσας

Είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και προτάσεων της γλώσσας. Σε μία γλώσσα προγραμματισμού αυτό καθορίζεται από το δημιουργό της ενώ σε μία φυσική γλώσσα από αυτόν που εκφέρει την πρόταση.

Διαφορές μεταξύ φυσικών και τεχνητών γλωσσών

Φυσικές	Τεχνητές
Χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπων.	Χρησιμοποιούνται για την επικοινωνία μεταξύ ανθρώπου και Η/Υ.
Έχουν μεγάλες δυνατότητες εξέλιξης. Νέες λέξεις μπορεί να εισαχθούν, κανόνες γραμματικής και σύνταξης να αλλάξουν κλπ.	Οι δυνατότητες εξέλιξης είναι περιορισμένες. Τις περισσότερες φορές η εξέλιξη αυτή αφορά την επέκταση του ρεπερτορίου των εντολών της γλώσσας (π.χ. Basic και VisualBasic).

6.4 Τεχνικές σχεδίασης προγραμμάτων

Δεν αρκεί κάποιος να γνωρίζει απλά μία γλώσσα προγραμματισμού. Θα πρέπει να ακολουθήσει και μία τεχνική σχεδίασης του προγράμματός του.

Για την σύνταξη σωστών, κατανοητών και εύκολα συντηρήσιμων προγραμμάτων ακολουθήθηκαν διάφορες μεθοδολογίες ανάπτυξης που παρουσιάζονται παρακάτω:

6.4.1 Ιεραρχική σχεδίαση ή Top down σχεδίαση

Η τεχνική αυτή συνιστά στον καθορισμό των βασικών λειτουργιών του προγράμματος σε ανώτερο επίπεδο και στη συνέχεια τη διάσπαση καθεμιάς σε μικρότερες και απλούστερες μέχρι του σημείου να είναι τόσο απλές που μπορούν να επιλυθούν άμεσα.

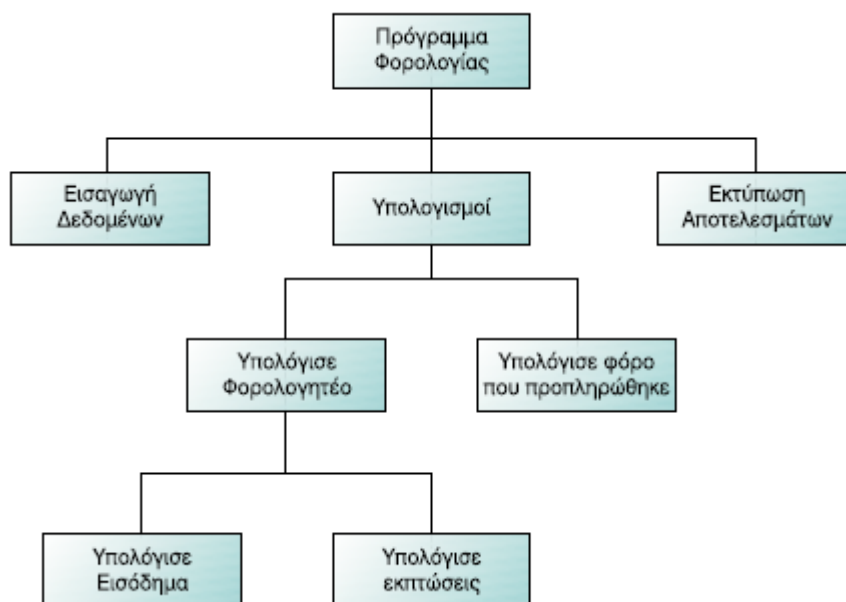
Συνεπώς, σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση του προβλήματος σε μικρότερα και απλούστερα υποπροβλήματα τα οποία είναι ευκολότερο να επιλυθούν.

Η ιεραρχική σχεδίαση ή ιεραρχικός προγραμματισμός χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα.

Η σωστή διαίρεση του αρχικού προβλήματος σε υποπροβλήματα και κατά συνέπεια του αρχικού προγράμματος σε τμήματα προγράμματος είναι μία διαδικασία αρκετά πολύπλοκη.

Για την Top-down σχεδίαση χρησιμοποιούμε το **ιεραρχικό διάγραμμα**.

Παράδειγμα: Πρόγραμμα Υπολογισμός Καθαρού Μισθού



6.4.2 Τμηματικός προγραμματισμός

Η ιεραρχική σχεδίαση υλοποιείται με τον τμηματικό προγραμματισμό.

Μετά την ανάλυση του προβλήματος σε μικρότερα και απλούστερα υποπροβλήματα, κάθε υποπρόβλημα αποτελεί μία ξεχωριστή και ανεξάρτητη **ενότητα** (module).

Τώρα, για κάθε ενότητα θα γραφτεί το κατάλληλο πρόγραμμα ή τμήμα προγράμματος.

6.4.3 Δομημένος προγραμματισμός

Είναι η μεθοδολογία που έχει επικρατήσει σήμερα. Εμπεριέχει τις αρχές της ιεραρχικής σχεδίασης και του τμηματικού προγραμματισμού.

Επιπλέον, αναφέρει ότι για τη δημιουργία σωστών προγραμμάτων να χρησιμοποιούμε μόνο τις 3 στοιχειώδεις δομές : **Ακολουθίας, Επιλογής και Επανάληψης**.

Όλα τα προγράμματα, οσοδήποτε μεγέθους, μπορούν να γραφτούν στηριζόμενα στη χρήση μόνο αυτών των δομών, αποφεύγοντας πλήρως τη χρήση τη δομής **GOTO**.

Επίσης, κάθε ενότητα προγράμματος έχει μόνο μία είσοδο και μόνο μία έξοδο.

Πλεονεκτήματα του δομημένου προγραμματισμού

- Άμεση υλοποίηση των αλγορίθμων σε πρόγραμμα.
- Διευκόλυνση της ανάλυσης του προγράμματος σε τμήματα (ενότητες).
Το κάθε τμήμα μπορεί να γραφτεί από διαφορετικές ομάδες προγραμματιστών.
- Ευκολότερη και συντομότερη ανάπτυξη προγραμμάτων.
- Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.
- Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.
- Ευκολότερη διόρθωση και συντήρηση του προγράμματος.

ΕΝΤΟΛΗ GOTO: παράδειγμα προς αποφυγή, το μαύρο πρόβατο τού προγραμματισμού

Στα πρώτα χρόνια δημιουργίας Γλωσσών Προγραμματισμού κυριαρχούσε ο **αδόμητος προγραμματισμός** και κυρίως η **εντολή διακλάδωσης GOTO** που ευθύνεται γι αυτόν και θεωρείται το **μαύρο πρόβατο** του προγραμματισμού.

Η εντολή GOTO θεωρείται επιβλαβής και θεμελιώνει το δομημένο προγραμματισμό. Χρειάστηκε όμως να περάσουν αρκετά χρόνια, ώστε να αρχίσει να διαδίδεται η χρήση του δομημένου προγραμματισμού.

Βασικός λόγος για τα προβλήματα αυτά ήταν η αλόγιστη χρήση μίας εντολής, της εντολής GOTO που χρησιμοποιούμενη άλλαζε διαρκώς τη ροή του προγράμματος. Ο δομημένος προγραμματισμός αναπτύχθηκε από την ανάγκη να υπάρχει μία κοινή μεθοδολογία στην ανάπτυξη των προγραμμάτων και τη μείωση των εντολών GOTO που χρησιμοποιούνται στο πρόγραμμα.

Η χρήση αυτής της εντολής είχε τα ακόλουθα **αποτελέσματα**:

1. **μεγάλα προγράμματα και ιδιαίτερα μπερδεμένα,**
2. **απαιτούνταν πάρα πολύς χρόνος για τη συγγραφή, διόρθωση και συντήρηση.**

Όλες οι **σύγχρονες γλώσσες υποστηρίζουν το δομημένο προγραμματισμό** και διαθέτουν εντολές που καθιστούν τη χρήση της εντολής **GOTO** περιττή.

6.5 Αντικειμενοστραφής προγραμματισμός

Πρόκειται για μια νέα τάση αντιμετώπισης προγραμματιστικών αντιλήψεων και δομών που ονομάζεται αντικειμενοστραφής προγραμματισμός. Την τελευταία δεκαετία έχει γίνει η επικρατούσα κατάσταση και έχει αλλάξει ριζικά τα μέχρι πριν από λίγα χρόνια γνωστά και σταθερά σημεία αναφοράς των προγραμματιστών.

Η ιδέα του αντικειμενοστραφούς προγραμματισμού ή της αντικειμενοστραφούς σχεδίασης έχει τις ρίζες της σε πολύ απλοϊκή ιδέα.

Ένα πρόγραμμα περιγράφει "ενέργειες" (επεξεργασία) που εφαρμόζονται πάνω σε δεδομένα. Ένα βασικό ερώτημα που τίθεται είναι αν η φιλοσοφία, η δομή του προγράμματος είναι προτιμότερο να στηρίζεται στις "ενέργειες" ή στα δεδομένα.

Η απάντηση σε αυτό το ερώτημα προσδιορίζει και τη βασική διαφορά ανάμεσα στις παραδοσιακές προγραμματιστικές τεχνικές και στην αντικειμενοστραφή προσέγγιση.

Η αντικειμενοστραφής σχεδίαση εκλαμβάνει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία δημιουργούνται με κατάλληλη μορφοποίηση τα αντικείμενα.

Αυτή η σχεδίαση αποδείχθηκε ότι επιφέρει καλύτερα αποτελέσματα, αφού τα προγράμματα που δημιουργούνται είναι περισσότερο ευέλικτα και επαναχρησιμοποιήσιμα.

Αντικειμενοστραφής προγραμματισμός (object-oriented programming) ή αντικειμενοστραφής σχεδίαση είναι μια μεθοδολογία ανάπτυξης εφαρμογών η οποία στηρίζεται σε αυτόνομες προγραμματιστικές οντότητες με δική τους ταυτότητα και συμπεριφορά.

Οι οντότητες αυτές καλούνται **αντικείμενα (objects)**, αντιστοιχούν σε φυσικές οντότητες ή έννοιες του φυσικού μας κόσμου, και δομούνται με βάση δεδομένα (**ιδιότητες**) που προσδιορίζουν την υπόστασή τους και ενέργειες (κανόνες συμπεριφοράς) που εφαρμόζονται πάνω στα δεδομένα. Σε μια εφαρμογή, ένα αντικείμενο είναι ο ομαδοποιημένος συνδυασμός δεδομένων και κώδικα, τα οποία έχουμε τη δυνατότητα να χειριστούμε ενιαία. Τα δεδομένα αποτελούν τα χαρακτηριστικά ενός αντικειμένου και αναφέρονται ως **ιδιότητες (properties)** ενώ οι ενέργειες καθορίζουν τη συμπεριφορά του. Οι ενέργειες στον αντικειμενοστραφή προγραμματισμό αναφέρονται και ως **μέθοδοι (methods)**.

Σύμφωνα με την αντικειμενοστραφή θεωρία ανάπτυξης εφαρμογών, η προσέγγιση κάθε προβλήματος πρέπει να γίνεται με φυσική ερμηνεία και να μη στηρίζεται σε πολύπλοκα τεχνικά ζητήματα. Αυτή η αντίληψη, ότι δηλαδή η επίλυση ενός προβλήματος επιτυγχάνεται με τη σύνθεση ικανοτήτων (ο τρόπος υλοποίησης των οποίων μας είναι άγνωστος) που διαθέτουν

διαφορετικές ανεξάρτητες οντότητες, οι οποίες αλληλεπιδρούν για τον σκοπό αυτό, βρίσκεται στην καρδιά της αντικειμενοστραφούς προσέγγισης.

Οι θεμελιώδεις αρχές του αντικειμενοστραφούς προγραμματισμού πηγάζουν από τον καθημερινό μας φυσικό κόσμο, καθώς είναι πολύ κοντά στον τρόπο που σκεφτόμαστε για να επιλύσουμε προβλήματα της καθημερινότητάς μας. Στηρίζονται στο γεγονός ότι για να μπορέσει κάποιος να

κατανοήσει άγνωστες σε αυτόν έννοιες, θα πρέπει να καθοδηγηθεί μέσω της προσομοίωσης των άγνωστων αυτών εννοιών αντιστοιχίζοντας αυτές σε πρακτικές γνώσεις και εικόνες από το περιβάλλον του, τις οποίες γνωρίζει και μπορεί πολύ εύκολα να χειριστεί.

Χτίζοντας Αντικειμενοστραφή Προγράμματα

Το «χτίσιμο» μιας αντικειμενοστραφούς εφαρμογής επιτυγχάνεται με τη δημιουργία και τον χειρισμό αντικειμένων τα οποία πρέπει να συνεργαστούν για την επίτευξη του κοινού στόχου που είναι η επίλυση του προβλήματος. Με ποιον όμως τρόπο εργαζόμαστε, ώστε να εντοπίσουμε τα απαραίτητα δομικά στοιχεία της εφαρμογής;

Βέβαια, δημιουργούνται μία σειρά από εύλογα ερωτήματα, όπως:

- "Τι ακριβώς είναι ένα αντικείμενο;"
- "Πώς προσδιορίζουμε και περιγράφουμε ένα αντικείμενο;"
- "Πώς το πρόγραμμα χειρίζεται τα αντικείμενα;"
- "Πώς τα αντικείμενα συσχετίζονται μεταξύ τους;"

Φυσικά ο αντικειμενοστραφής προγραμματισμός χρησιμοποιεί την ιεραρχική σχεδίαση, τον τμηματικό προγραμματισμό και ακολουθεί τις αρχές του δομημένου προγραμματισμού.

Μεθοδολογία

Το μόνο που έχουμε να κάνουμε είναι να αναλύσουμε το πρόβλημα το οποίο θέλουμε να επιλύσουμε, δηλαδή να αναγνωρίσουμε και να καταγράψουμε τα βασικά συστατικά στοιχεία της διαδικασίας επίλυσής του που είναι:

1. τα αντικείμενα που συμμετέχουν με βάση τον ρόλο τους στο συγκεκριμένο σενάριο,
2. οι ιδιότητες κάθε αντικειμένου, δηλ. τα σχετικά με το συγκεκριμένο πρόβλημα χαρακτηριστικά του, και
3. οι υπηρεσίες που προσφέρει ή οι ενέργειες που υλοποιεί κάθε αντικείμενο (μέθοδοι) προς αξιοποίηση από άλλες, ώστε να αναπτυχθούν οι απαραίτητες συνεργασίες μεταξύ των αντικειμένων για την επίλυση του προβλήματος.

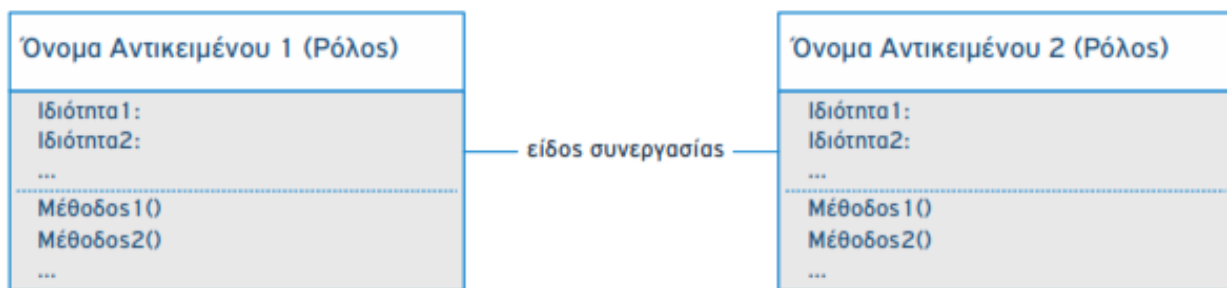
Πρόβλημα «Αποστολή λουλουδιών» (συστατικά επίλυσης προβλήματος)

Με βάση την παραπάνω μεθοδολογία είμαστε ήδη σε θέση να αρχίσουμε να αποτυπώνουμε το αντικειμενοστραφές περιβάλλον και τα συστατικά επίλυσης του προβλήματος «Αποστολή λουλουδιών» που αποτελεί τη βάση ανάπτυξης μιας πραγματικής αντικειμενοστραφούς εφαρμογής. Η εφαρμογή αυτή θα έδινε στη μαμά τη δυνατότητα ηλεκτρονικής παραγγελίας και αποστολής λουλουδιών από τα ηλεκτρονικά καταστήματα των ανθοπωλών μας!

Διαγραμματική αναπαράσταση

Αφού εντοπίσουμε τα συστατικά επίλυσης του προβλήματος, μπορούμε να τα οργανώσουμε σε μια απλή διαγραμματική αναπαράσταση χρησιμοποιώντας παραλληλόγραμμα για την αποτύπωση των αντικειμένων, των ιδιοτήτων και των μεθόδων τους και γραμμές σύνδεσης για την περιγραφή του είδους της μεταξύ τους συνεργασίας.

Η αναπαράσταση αυτή είναι ιδιαίτερα σημαντική, διότι μας δίνει την εποπτική εικόνα των συνεργαζόμενων οντοτήτων του προβλήματός μας και ουσιαστικά αποτελεί το σχέδιο επίλυσής του με βάση την αντικειμενοστραφή προσέγγιση.



Ένα **αντικειμενοστραφές πρόγραμμα δομείται** ως ένα δίκτυο συνεργαζόμενων οντοτήτων που είναι τα αντικείμενα. Κάθε αντικείμενο έχει ένα συγκεκριμένο ρόλο στην εφαρμογή και παρέχει μια υπηρεσία ή εκτελεί μια ενέργεια (μέθοδο) που χρησιμοποιείται από άλλα μέλη του δικτύου, δηλαδή από άλλα αντικείμενα, για την υλοποίηση της συνεργασίας που θα επιλύσει το πρόβλημα.

Ομαδοποίηση Αντικειμένων σε Κλάσεις: Αφαιρετικότητα και Ενθυλάκωση

Κατά τη διαδικασία οργάνωσης και διαγραμματικής αναπαράστασης του προβλήματος της αποστολής λουλουδιών είδαμε ότι κάθε αντικείμενο περιέχει ένα σύνολο ιδιοτήτων και μεθόδων που ενεργούν πάνω στο αντικείμενο. Όπως ήδη σχολιάσαμε, στόχος αυτής της δόμησης των αντικειμένων είναι η απόκρυψη των λεπτομερειών υλοποίησης και λειτουργίας τους από τον υπόλοιπο κόσμο. Ουσιαστικά το αντικείμενο αποτελεί έναν «θύλακα», δηλαδή ένα σακούλι στο οποίο αποθηκεύει και συνδυάζει τα δεδομένα (ιδιότητες) και τις λειτουργίες (μεθόδους) του. Λέμε λοιπόν ότι τα αντικείμενα παρέχουν έναν τρόπο ενθυλάκωσης δεδομένων και λειτουργιών σε αυτά.

Σε μια αντικειμενοστραφή εφαρμογή κάθε αντικείμενο **αποτελεί ξεχωριστή οντότητα** και περιέχει **ενσωματωμένες τις ιδιότητες** (δεδομένα) και **τους κανόνες συμπεριφοράς του** (μεθόδους).

Η δυνατότητα ενός αντικειμένου να συνδυάζει εσωτερικά τα δεδομένα και τις μεθόδους χειρισμού του καλείται ενθυλάκωση (encapsulation).

Την ενθυλάκωση μπορούμε να την παρομοιάσουμε σαν ένα κέλυφος που υπάρχει γύρω από κάθε αντικείμενο και διαχωρίζει τον εσωτερικό από τον εξωτερικό του κόσμο.

Για οικονομία στην ανάπτυξη αντικειμενοστραφών προγραμμάτων ομαδοποιούμε τα αντικείμενα σε κλάσεις.

Ο γενικός τύπος ενός αντικειμένου καλείται κλάση (class) και καθορίζει τις αρχικές ιδιότητες και τη συμπεριφορά κάθε αντικειμένου που προέρχεται από αυτή. Μια κλάση αποτελεί ένα αφαιρετικό (abstract) στοιχείο (τύπο) και μπορεί να παράγει ένα απεριόριστο πλήθος δομικά ίδιων αντικειμένων.

Διαγραμματική Αναπαράστασης Κλάσεων

Για την υλοποίηση της διαγραμματικής αναπαράστασης απαιτείται η πλήρης προδιαγραφή καθεμιάς από τις συμμετέχουσες κλάσεις, δηλαδή ο εντοπισμός των ιδιοτήτων, των μεθόδων και του είδους της μεταξύ τους συνεργασίας.

Οι **ιδιότητες** των κλάσεων που απορρέουν από τις πληροφορίες του ενός σεναρίου είναι:

1. Πελάτης: Όνομα, Επώνυμο, Κωδικός, Διεύθυνση, Τηλέφωνο
2. Κατάστημα: Ονομασία, Διεύθυνση
3. Ταινία: Τίτλος, Αριθμός

Τα είδη συνεργασίας που περιγράφονται στο σενάριο μεταξύ των συμμετεχόντων αντικειμένων είναι:

1. Είναι πελάτης: Γιώργος Πέτρου (Πελάτης) - ΒιντεοΔράση (Κατάστημα)
2. Ενοικιάζει: Γιώργος Πέτρου (Πελάτης) - Πόλεμος των Άστρων (Ταινία)
3. Αποθηκεύει: ΒιντεοΔράση (Κατάστημα) - Πόλεμος των Άστρων (Ταινία)

Με βάση τα παραπάνω είδη συνεργασίας, οι ελάχιστες ενέργειες που πρέπει να υποστηρίζουν τα αντικείμενα, δηλαδή οι **μέθοδοι** που πρέπει να υλοποιούν είναι:

1. Γιώργος Πέτρου (Πελάτης): ΖητώΕίσοδο(), ΖητώΕνοικίαση()
2. ΒιντεοΔράση (Κατάστημα): ΤαυτοποιώΠελάτη(), ΑποθηκεύωΤαινία()
3. Πόλεμος των Άστρων (Ταινία): Ενοικίαση()

6.6 Παράλληλος προγραμματισμός

Μία άλλη μορφή προγραμματισμού που αναπτύσσεται τελευταία και πιθανόν στο μέλλον να γνωρίσει μεγάλη άνθηση είναι ο παράλληλος προγραμματισμός. Σχετικά πρόσφατα εμφανίστηκαν υπολογιστές που ξεφεύγουν από την κλασική αρχιτεκτονική και διαθέτουν περισσότερους από έναν επεξεργαστές.

Οι επεξεργαστές αυτοί μοιράζονται την ίδια μνήμη και λειτουργούν παράλληλα εκτελώντας διαφορετικές εντολές του ιδίου προγράμματος. Οι υπολογιστές αυτοί εμφανίζονται θεωρητικά να πετυχαίνουν ταχύτητες, που είναι ασύλληπτες για τους τυπικούς υπολογιστές με έναν επεξεργαστή.

Για να εκμεταλλευτούμε όμως την ταχύτητα που προσφέρει η αρχιτεκτονική τους, πρέπει το πρόβλημα να διαιρεθεί σε τμήματα που εκτελούνται παράλληλα και στη συνέχεια να

προγραμματιστεί σε ένα προγραμματιστικό περιβάλλον που να επιτρέπει τον παράλληλο προγραμματισμό.

Για λόγους συμβατότητας και μόνο, μερικές την διατηρούν ακόμα στο ρεπερτόριο των εντολών τους.

6.7 Προγραμματιστικά περιβάλλοντα

Ένα πρόγραμμα που φτιάχνεται σε μία γλώσσα υψηλού επιπέδου δεν είναι άμεσα κατανοητό από τον Η/Υ. Θα πρέπει να μεταφραστεί σε ισοδύναμο πρόγραμμα σε γλώσσα μηχανής (δυαδική μορφή). Την διαδικασία μετάφρασης την πραγματοποιούν τα μεταφραστικά προγράμματα. Είναι δύο ειδών:

- **Μεταγλωττιστές (Compilers)**
- **Διερμηνευτές (Interpreters)**

Συντάκτης: Ειδικό πρόγραμμα για την σύνταξη των προγραμμάτων σε γλώσσα υψηλού επιπέδου δηλαδή του πηγαίου προγράμματος.

Πηγαίο πρόγραμμα : Το πρόγραμμα που είναι φτιαγμένο σε γλώσσα υψηλού επιπέδου.

Μεταγλωττιστής: Παίρνει, ως είσοδο, το πηγαίο πρόγραμμα και αναλαμβάνει να το μεταφράσει εξ' ολοκλήρου παράγοντας το αντικείμενο πρόγραμμα.

Διερμηνευτής: Παίρνει, ως είσοδο, μία-μία εντολή του πηγαίου, την μεταφράζει και την εκτελεί αμέσως. Η λειτουργία του μοιάζει με τον άνθρωπο-διερμηνέα που μεταφράζει επί τόπου κάθε πρόταση.

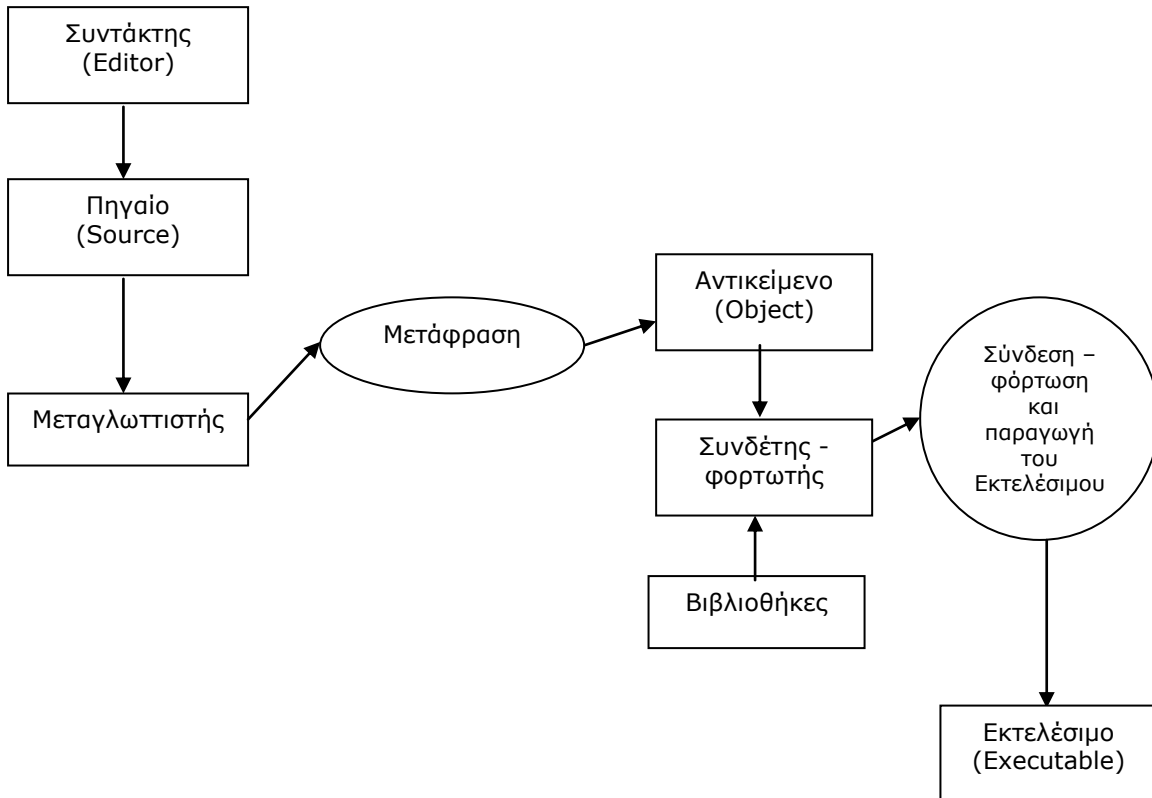
Αντικείμενο πρόγραμμα (Object program): Το πρόγραμμα που είναι μεταφρασμένο σε γλώσσα μηχανής αλλά όχι άμεσα εκτελέσιμο.

Βιβλιοθήκες (Libraries): Έτοιμες ενότητες (modules) αντικείμενου προγράμματος της γλώσσας, απαραίτητες για την παραγωγή του εκτελέσιμου προγράμματος.

Συνδέτης – Φορτωτής (Linker-Loader): Ειδικό πρόγραμμα που αναλαμβάνει να συνδέσει το αντικείμενο πρόγραμμα με τις βιβλιοθήκες και να παράγει το εκτελέσιμο.

Εκτελέσιμο πρόγραμμα (Executable program): Το πρόγραμμα σε γλώσσα μηχανής που είναι έτοιμο πλέον να εκτελεστεί.

Η διαδικασία μεταγλώττισης και σύνδεσης προγράμματος φαίνεται παρακάτω:



Ο μεταγλωττιστής ή ο διερμηνευτής ανιχνεύει λοιπόν τα **συντακτικά λάθη** και εμφανίζει κατάλληλα διαγνωστικά μηνύματα.

Το στάδιο που ακολουθεί είναι η διόρθωση των λαθών.

Το διορθωμένο πρόγραμμα επαναυποβάλεται για μεταγλώττιση και η διαδικασία αυτή επαναλαμβάνεται, μέχρις ότου εξαλειφθούν πλήρως όλα τα λάθη.

Η χρήση μεταγλωττιστή έχει το **ΜΕΙΟΝΕΚΤΗΜΑ** ότι, προτού χρησιμοποιηθεί ένα πρόγραμμα, πρέπει να περάσει από τη διαδικασία της **ΜΕΤΑΓΛΩΤΤΙΣΗΣ** και **ΣΥΝΔΕΣΗΣ**.

Από την άλλη μεριά η χρήση διερμηνευτή έχει το **ΠΛΕΟΝΕΚΤΗΜΑ** της **ΆΜΕΣΗΣ** εκτέλεσης και συνεπώς και της **ΆΜΕΣΗΣ** διόρθωσης.

Όμως η εκτέλεση του προγράμματος μέσω του διερμηνευτή καθίσταται πιο αργή, σημαντικά μερικές φορές, από εκείνη του ισοδύναμου εκτελέσιμου προγράμματος που παράγει ο μεταγλωττιστής.

Πάντως τα σύγχρονα προγραμματιστικά περιβάλλοντα παρουσιάζονται συνήθως με μεικτές υλοποιήσεις, όπου χρησιμοποιείται διερμηνευτής κατά τη φάση δημιουργίας του προγράμματος και μεταγλωττιστής για την τελική έκδοση και εκμετάλλευση του προγράμματος.

Κατά τη δημιουργία ενός προγράμματος σχεδόν πάντα ενυπάρχουν λάθη. Τα λάθη τα χωρίζουμε σε δύο κατηγορίες :

- **Συντακτικά λάθη**
- **Λογικά λάθη**

Τα **συντακτικά λάθη** ανιχνεύονται κατά την διαδικασία της μεταγλώττισης ή διερμηνεύσης.

Αφορούν παραβιάσεις του τυπολογικού και συντακτικού της γλώσσας. (π.χ. μία εντολή έχει γραφτεί συντακτικά λάθος).

Στην περίπτωση αυτή ο προγραμματιστής πρέπει να επιστρέψει στο πηγαίο πρόγραμμα, να διορθώσει τα λάθη και να το ξανά υποβάλλει για μεταγλώττιση.

Τα **λογικά λάθη** είναι και τα πλέον δύσκολα στην ανίχνευσή τους.

Έχουν να κάνουν με σφάλματα στη λογική επίλυσης του προβλήματος ή λανθασμένης διατύπωσης του αλγορίθμου (π.χ. το πρόγραμμα παράγει άλλα αποτελέσματα κι όχι τα ζητούμενα!).

Για την σύνταξη των προγραμμάτων χρησιμοποιούμε ένα ειδικό πρόγραμμα που ονομάζεται **συντάκτης (editor)**.

Μοιάζει με επεξεργαστή κειμένου με επιπλέον δυνατότητες που διευκολύνουν την γρήγορη σύνταξη των πηγαίων προγραμμάτων.

Τα σύγχρονα προγραμματιστικά περιβάλλοντα περιέχουν οπωσδήποτε 3 ειδών προγράμματα : Τον συντάκτη, το μεταγλωττιστή και το συνδέτη-φορτωτή. Πέραν αυτών, περιέχουν όλα εκείνα τα εργαλεία που διευκολύνουν την εύκολη, ταχύτατη σύνταξη, διόρθωση και συντήρηση των προγραμμάτων γι' αυτό και πολλές φορές ονομάζονται ως **RAD περιβάλλοντα (Rapid Application Development)**.

Όλα αυτά παρέχονται με έναν ενιαίο και λειτουργικό τρόπο στον προγραμματιστή.

GOTO

ΕΝΤΟΛΗ GOTO: παράδειγμα προς αποφυγή, το μαύρο πρόβατο τού προγραμματισμού.

Στα πρώτα χρόνια δημιουργίας Γλωσσών Προγραμματισμού κυριαρχούσε ο αδόμητος προγραμματισμός και κυρίως η εντολή διακλάδωσης GOTO που ευθύνεται γι αυτόν και θεωρείται το μαύρο πρόβατο του.

Η χρήση αυτής της εντολής είχε τα ακόλουθα **αποτελέσματα**:

1. **μεγάλα προγράμματα και ιδιαίτερα μπερδεμένα,**
2. **απαιτούνταν πάρα πολύς χρόνος για τη συγγραφή, διόρθωση και συντήρηση.**

Όλες οι **σύγχρονες γλώσσες υποστηρίζουν το δομημένο προγραμματισμό** και διαθέτουν εντολές που καθιστούν τη χρήση της εντολής GOTO περιττή. Για λόγους συμβατότητας και μόνο, μερικές την διατηρούν ακόμα στο ρεπερτόριο των εντολών τους.

ΚΑΝΟΝΕΣ ΜΕΤΑΤΡΟΠΗΣ GOTO ΣΕ ΔΟΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

1^{ος} ΚΑΝΟΝΑΣ	
ΕΤΙΚΕΤΑ 1:ΕΝΤΟΛΗ_Α } ΕΝΤΟΛΕΣ ΑΝ ΣΥΝΘΗΚΗ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ 1 ΕΝΤΟΛΗ_Β	ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΕΝΤΟΛΗ_Α } ΕΝΤΟΛΕΣ ΜΕΧΡΙΣ_ΟΤΟΥ ΟΧΙ(ΣΥΝΘΗΚΗ) ΕΝΤΟΛΗ_Β
2^{ος} ΚΑΝΟΝΑΣ	
ΕΤΙΚΕΤΑ 1: ΑΝ ΣΥΝΘΗΚΗ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ 2 } ΕΝΤΟΛΕΣ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ 1 ΕΤΙΚΕΤΑ 2: ΕΝΤΟΛΗ_Β	ΟΣΟ ΟΧΙ(ΣΥΝΘΗΚΗ) ΕΠΑΝΑΛΑΒΕ ΕΝΤΟΛΗ_Α } ΕΝΤΟΛΕΣ ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΕΝΤΟΛΗ_Β
3^{ος} ΚΑΝΟΝΑΣ	
ΑΝ ΣΥΝΘΗΚΗ ΤΟΤΕ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ 1 ΕΝΤΟΛΕΣ_Α ΕΤΙΚΕΤΑ 1: ΕΝΤΟΛΕΣ_Β	ΑΝ ΟΧΙ(ΣΥΝΘΗΚΗ) ΤΟΤΕ ΕΝΤΟΛΕΣ_Β ΤΕΛΟΣ_ΑΝ ΕΝΤΟΛΕΣ_Α
4^{ος} ΚΑΝΟΝΑΣ	
ΑΝ ΣΥΝΘΗΚΗ ΤΟΤΕ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ 1 ΕΝΤΟΛΕΣ_Α ΕΤΙΚΕΤΑ 1: ΕΝΤΟΛΕΣ_Β	ΑΝ ΟΧΙ(ΣΥΝΘΗΚΗ) ΤΟΤΕ ΕΝΤΟΛΕΣ_Β ΑΛΛΙΩΣ ΕΝΤΟΛΕΣ_Α ΤΕΛΟΣ_ΑΝ
5^{ος} ΚΑΝΟΝΑΣ	
ΑΝ ΣΥΝΘ1 ΤΟΤΕ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ1 ΑΝ ΣΥΝΘ2 ΤΟΤΕ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ2 ΕΝΤΟΛΗ_Γ GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ3 ΕΤΙΚΕΤΑ1: ΕΝΤΟΛΗ_Α GOTO (ΠΗΓΑΙΝΕ) ΕΤΙΚΕΤΑ3 ΕΤΙΚΕΤΑ2: ΕΝΤΟΛΗ_Β ΕΤΙΚΕΤΑ3: ΕΝΤΟΛΗ_Δ	ΑΝ ΣΥΝΘ1 ΤΟΤΕ ΕΝΤΟΛΕΣ_Α ΑΛΛΙΩΣ_ΑΝ ΣΥΝΘ2 ΤΟΤΕ ΕΝΤΟΛΕΣ_Β ΑΛΛΙΩΣ ΕΝΤΟΛΕΣ_Γ ΤΕΛΟΣ_ΑΝ ΕΝΤΟΛΗ_Δ

ΑΣΚΗΣΕΙΣ ΔΟΜΗΜΕΝΟΥ ΚΑΙ ΜΗ ΔΟΜΗΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Να μετατρέψετε τον αλγόριθμο που χρησιμοποιεί την εντολή goto (πήγαινε) σε αλγόριθμο που ακολουθεί τις αρχές του δομημένου προγραμματισμού και να σχεδιάσετε το διάγραμμα ροής.

Άσκηση 1

Αλγόριθμος Μη_δομημένο_1

ΣημείοΑ: Διάβασε x

Αν $(x \bmod 2 = 1)$ **τότε**

$counter1 \leftarrow counter1 + 1$

αλλιώς

$counter2 \leftarrow counter2 + 1$

Τέλος_αν

Αν $(x = 99)$ **τότε**

πήγαινε στο **ΣημείοΒ**

αλλιώς

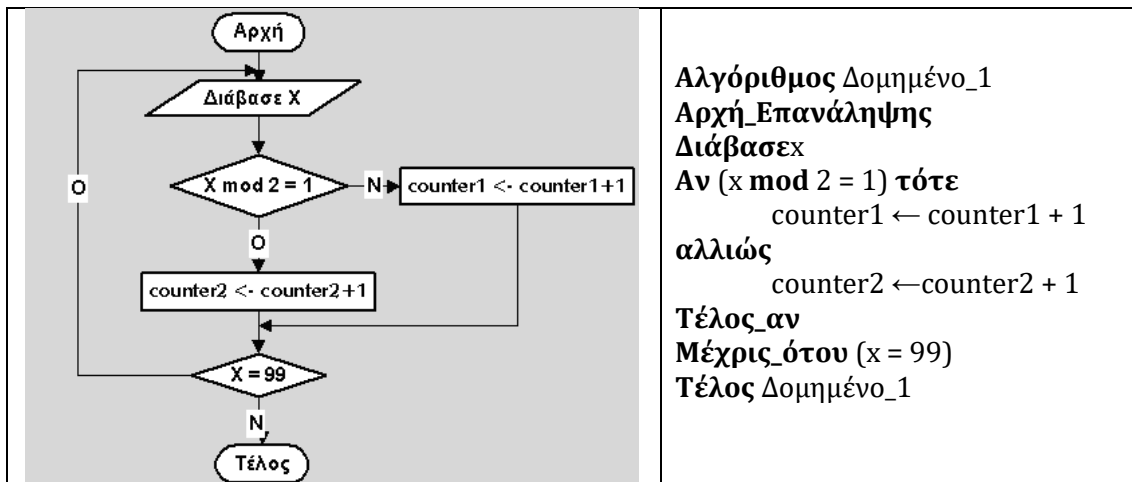
πήγαινε στην **ΣημείοΑ**

Τέλος_αν

ΣημείοΒ:

Τέλος Μη_δομημένο_1

Λύση



Άσκηση 2

Αλγόριθμος Μη_δομημένο_2

Όσο συνθήκη1 **επανάλαβε**

Εντολή1

Αν συνθήκη2 **τότε**

Εντολή3

Αλλιώς

Εντολή2

Πήγαινε στο **ΣημείοΑ**

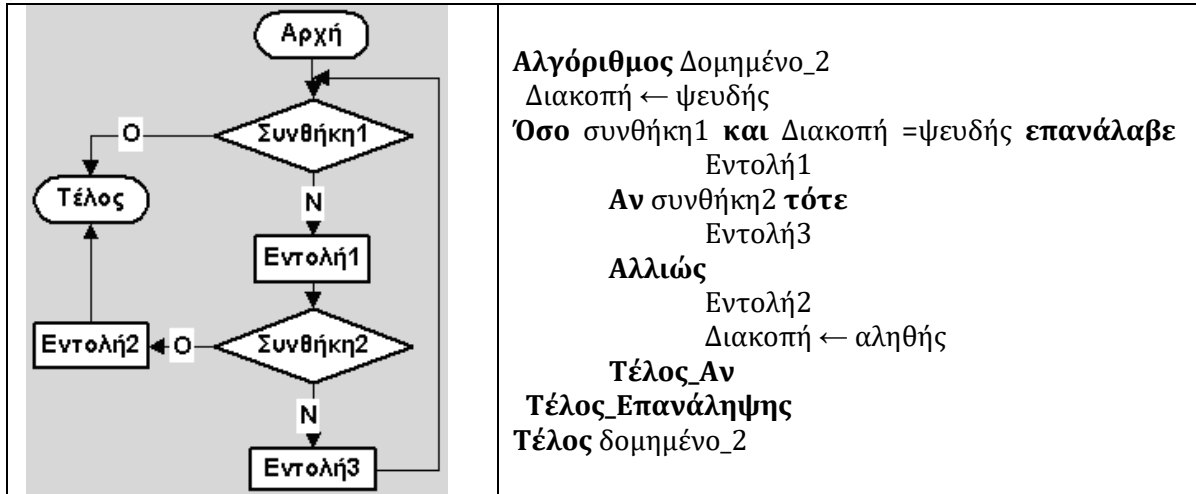
Τέλος_Αν

Τέλος_Επανάληψης

ΣημείοΑ:

Τέλος Μη_δομημένο_2

Λύση



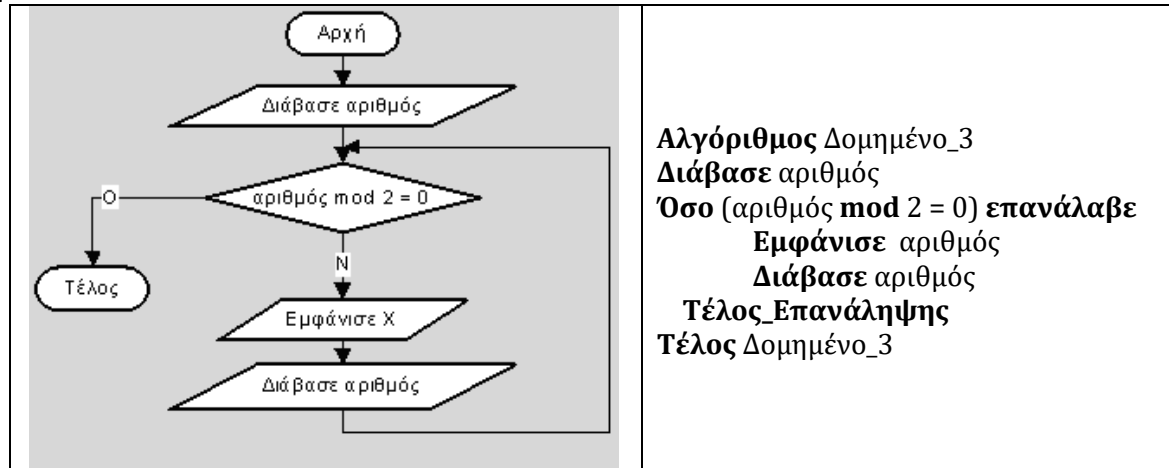
Άσκηση 3

Αλγόριθμος Μη_δομημένο_3

ΣημείοΑ:Διάβασε αριθμός
Αν (αριθμός mod 2 = 0) **τότε**
 Εμφάνισε αριθμός
 πήγαινε στο ΣημείοΑ
Τέλος_Αν

Τέλος Μη_δομημένο_3

Λύση



ΛΥΜΕΝΕΣ ΑΣΚΗΣΕΙΣ ΣΤΗΝ GOTO

Άσκηση 1

Ξαναγράψτε το παρακάτω σύνολο εντολών, έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
10 : ΔΙΑΒΑΣΕ Χ
    ΓΡΑΨΕ Χ^2
    ΑΝ Χ > 0 ΤΟΤΕ GOTO 20
    ΓΡΑΨΕ Χ+1
20: ΓΡΑΨΕ Χ-1
    ΔΙΑΒΑΣΕ Υ
    ΑΝ Υ <> Χ ΤΟΤΕ GOTO 10
    ΓΡΑΨΕ Α_Τ(Χ-1000)
```

Λύση

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Χ
ΓΡΑΨΕ Χ^2
ΑΝ Χ > 0 ΤΟΤΕ
    ΓΡΑΨΕ Χ-1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ Χ+1
ΤΕΛΟΣ_ΑΝ
ΔΙΑΒΑΣΕ Υ
ΜΕΧΡΙΣ_ΟΤΟΥ Υ = Χ
ΓΡΑΨΕ Α_Τ(Χ-1000)
```

Άσκηση 2

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
ΔΙΑΒΑΣΕ Χ
5: ΑΝ Χ <= 0 ΤΟΤΕ GOTO 10
    ΓΡΑΨΕ Χ
    ΔΙΑΒΑΣΕ Χ
    GOTO 5
10: ΓΡΑΨΕ 'ΤΕΛΟΣ'
```

Λύση

```
ΔΙΑΒΑΣΕ Χ
ΟΣΟ Χ > 0 ΕΠΑΝΑΛΑΒΕ
    ΓΡΑΨΕ Χ
    ΔΙΑΒΑΣΕ Χ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'ΤΕΛΟΣ'
```

Άσκηση 3

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
10 : ΔΙΑΒΑΣΕ ΕΤΟΣ
    ΓΡΑΨΕ 'ΕΥΤΥΧΙΣΜΕΝΟ ΤΟ', ΕΤΟΣ
    ΑΝ ΕΤΟΣ <> 2017 ΤΟΤΕ GOTO 10
```

Λύση

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ ΕΤΟΣ
ΓΡΑΨΕ 'ΕΥΤΥΧΙΣΜΕΝΟ ΤΟ', ΕΤΟΣ
ΜΕΧΡΙΣ_ΟΤΟΥ ΕΤΟΣ = 2017
```

Άσκηση 4

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
Α ← 0
10 : ΔΙΑΒΑΣΕ Χ
    ΑΝ Χ >= 20 ΤΟΤΕ GOTO 20
    Α ← Α+Χ^2
20: ΑΝ Α < 10^3 ΤΟΤΕ GOTO 10
    ΓΡΑΨΕ Α
```

Λύση

```
Α ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
ΔΙΑΒΑΣΕ Χ
ΑΝ Χ < 20 ΤΟΤΕ
    Α ← Α+Χ^2
ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ Α >= 10^3
ΓΡΑΨΕ Α
```

Άσκηση 5

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

```
A ← 0
K ← 100
10 : AN K < 0 ΤΟΤΕ GOTO 60
  Λ ← 4
20 : AN Λ > 80 ΤΟΤΕ GOTO 70
  ΓΡΑΨΕ K^2+Λ^2
  Λ ← Λ+4
70: GOTO 20
  K ← K-1
60: GOTO 10
  ΓΡΑΨΕ 'ΑΝΤΙΟ 2016'
```

Λύση

```
A ← 0
K ← 100
ΟΣΟ K >= 0 ΕΠΑΝΑΛΑΒΕ
  Λ ← 4
  ΟΣΟ Λ <= 80 ΕΠΑΝΑΛΑΒΕ
    ΓΡΑΨΕ K^2+Λ^2
    Λ ← Λ+4
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  K ← K-1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'ΑΝΤΙΟ 2016'
```

Άσκηση 6

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

```
ΔΙΑΒΑΣΕ Y
AN X < 100 ΤΟΤΕ GOTO 100
AN X MOD 5 <> 0 ΤΟΤΕ GOTO 100
Y ← Y+X
100 : ΓΡΑΨΕ Y
```

Λύση

```
ΔΙΑΒΑΣΕ Y
AN X >= 100 ΤΟΤΕ
  AN X MOD 5 = 0 ΤΟΤΕ
    Y ← Y+X
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ Y
```

Άσκηση 7

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
A ← 0
K ← 21
10: ΔΙΑΒΑΣΕ X
  A ← A+X*K
  K ← K+3
  AN A < 9999 ΤΟΤΕ GOTO 10
  ΓΡΑΨΕ A
```

Λύση

```
A ← 0
K ← 21
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ X
  A ← A+X*K
  K ← K+3
ΜΕΧΡΙΣ_ΟΤΟΥ A >= 9999
ΓΡΑΨΕ A
```

Άσκηση 8

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```

σ ← 0
μ ← 1
5: ΑΝ μ > 20 ΤΟΤΕ GOTO 15
α ← 1
10: ΔΙΑΒΑΣΕ λ
α ← α*λ
ΑΝ α < 1890 ΤΟΤΕ GOTO 10
σ ← σ+α
μ ← μ+1
GOTO 5
15: ΓΡΑΨΕ σ

```

Λύση

```

σ ← 0
μ ← 1
ΟΣΟ μ ≤ 20 ΕΠΑΝΑΛΑΒΕ
    α ← 1
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ λ
        α ← α*λ
    ΜΕΧΡΙΣ_ΟΤΟΥ α ≥ 1890
    σ ← σ+α
    μ ← μ+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ σ

```

Άσκηση 9

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```

Α ← 0
10: ΔΙΑΒΑΣΕ Χ
ΑΝ Χ ≤ 0 ΤΟΤΕ GOTO 10
Α ← Α+Χ
ΑΝ Α ≤ 10000 ΤΟΤΕ GOTO 10
ΑΝ Α < 15000 ΤΟΤΕ GOTO 20
ΑΝ Α < 25000 ΤΟΤΕ GOTO 30
ΓΡΑΨΕ 'γ'
GOTO 40
20: ΓΡΑΨΕ 'α'
GOTO 40
30: ΓΡΑΨΕ 'β'
40: ΓΡΑΨΕ 'ΤΕΛΟΣ'

```

Λύση

```

Α ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ΔΙΑΒΑΣΕ Χ
        ΜΕΧΡΙΣ_ΟΤΟΥ Χ > 0
        Α ← Α+Χ
    ΜΕΧΡΙΣ_ΟΤΟΥ Α > 10000
    ΑΝ Α < 15000 ΤΟΤΕ
        ΓΡΑΨΕ 'α'
    ΑΛΛΙΩΣ_ΑΝ Α < 25000 ΤΟΤΕ
        ΓΡΑΨΕ 'β'
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'γ'
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ 'ΤΕΛΟΣ'

```

Άσκηση 10

Επαναδιατυπώστε το παρακάτω σύνολο εντολών έτσι ώστε να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

```

Γ ← 1
ΔΙΑΒΑΣΕ Χ
10: ΑΝ Χ = 0 ΤΟΤΕ GOTO 20
Γ ← Γ*Χ
ΔΙΑΒΑΣΕ Χ
GOTO 10
20: ΓΡΑΨΕ Γ

```

Λύση

```

Γ ← 1
ΔΙΑΒΑΣΕ Χ
ΟΣΟ Χ <> 0 ΕΠΑΝΑΛΑΒΕ
    Γ ← Γ*Χ
    ΔΙΑΒΑΣΕ Χ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Γ

```

Άσκηση 11

Επαναδιατυπώστε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
A ← 0
10 : ΔΙΑΒΑΣΕ X, Y
    AN X > Y ΤΟΤΕ GOTO 25
    AN X < Y ΤΟΤΕ GOTO 30
    Z ← (X-Y)^2
    GOTO 35
25: Z ← X*Y+1
    GOTO 35
30: Z ← X+Y*2
35: ΓΡΑΨΕ Z
    AN X+Y < 1000 ΤΟΤΕ GOTO 10
    ΓΡΑΨΕ X^Y
```

Λύση

```
A ← 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ X, Y
    AN X > Y ΤΟΤΕ
        Z ← X*Y+1
    ΑΛΛΙΩΣ_ΑΝ X < Y ΤΟΤΕ
        Z ← X+Y*2
    ΑΛΛΙΩΣ
        Z ← (X-Y)^2
    ΤΕΛΟΣ_ΑΝ
    ΓΡΑΨΕ Z
ΜΕΧΡΙΣ_ΟΤΟΥ X+Y > = 1000
ΓΡΑΨΕ X^Y
```

Άσκηση 12

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ικανοποιεί τις αρχές του δομημένου προγραμματισμού.

```
ΔΙΑΒΑΣΕ Λ
AN Λ > 0 ΤΟΤΕ GOTO 10
Λ ← -Λ
10: α ← 1
    κ ← 1
20: AN κ > Λ ΤΟΤΕ GOTO 30
    α ← α*(κ+1)
    κ ← κ+1
    GOTO 20
30: ΓΡΑΨΕ α
```

Λύση

```
ΔΙΑΒΑΣΕ Λ
AN Λ <= 0 ΤΟΤΕ
    Λ ← -Λ
ΤΕΛΟΣ_ΑΝ
α ← 1
κ ← 1
ΟΣΟ κ <= Λ ΕΠΑΝΑΛΑΒΕ
    α ← α*(κ+1)
    κ ← κ+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ α
```

Άσκηση 13

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να διέπεται από τις αρχές του δομημένου προγραμματισμού.

```
μ ← 1
κ ← 1
10: ΔΙΑΒΑΣΕ α
    AN α > 0 ΤΟΤΕ GOTO 20
    α ← -α
20: μ ← μ*α
    κ ← κ+1
    AN κ <= 98 ΤΟΤΕ GOTO 10
    ΓΡΑΨΕ μ
```

Λύση

```
μ ← 1
κ ← 1
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ α
    AN α <= 0 ΤΟΤΕ
        α ← -α
    ΤΕΛΟΣ_ΑΝ
    μ ← μ*α
    κ ← κ+1
ΜΕΧΡΙΣ_ΟΤΟΥ κ > 98
ΓΡΑΨΕ μ
```

Άσκηση 14

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε αυτό να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

ΔΙΑΒΑΣΕ μ, ν

ΑΝ $\mu \leq \nu$ ΤΟΤΕ GOTO 11

$B \leftarrow \mu$

$\mu \leftarrow \nu$

$\nu \leftarrow B$

11: ΓΡΑΨΕ μ, ν

Λύση

ΔΙΑΒΑΣΕ μ, ν

ΑΝ $\mu > \nu$ ΤΟΤΕ

$B \leftarrow \mu$

$\mu \leftarrow \nu$

$\nu \leftarrow B$

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ μ, ν

Άσκηση 15

Ξαναγράψτε το παρακάτω σύνολο εντολών έτσι ώστε να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

$\mu \leftarrow 71$

10: ΔΙΑΒΑΣΕ λ

$\mu \leftarrow \mu - \lambda$

ΑΝ $\mu > 0$ ΤΟΤΕ GOTO 10

ΓΡΑΨΕ μ

Λύση

$\mu \leftarrow 71$

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ λ

$\mu \leftarrow \mu - \lambda$

ΜΕΧΡΙΣ_ΟΤΟΥ $\mu \leq 0$

ΓΡΑΨΕ μ

Άσκηση 16

Δίνεται το παρακάτω σύνολο εντολών:

3 : ΔΙΑΒΑΣΕ X

$AB \leftarrow 0$

ΑΝ $X < 0$ ΤΟΤΕ GOTO 4

ΑΝ $X < 19$ ΤΟΤΕ GOTO 5

$Y \leftarrow X - 1$

GOTO 6

4 : $Y \leftarrow A_T(X)$

GOTO 6

5 : $Y \leftarrow 20 * X$

6 : $B \leftarrow 1$

7 : ΑΝ $B > Y$ ΤΟΤΕ GOTO 8

ΔΙΑΒΑΣΕ Z

$AB \leftarrow AB + Z$

$B \leftarrow B + 1$

GOTO 7

8 : ΓΡΑΨΕ AB

ΑΝ $AB < 2356$ ΤΟΤΕ GOTO 3

Λύση

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ X

$AB \leftarrow 0$

ΑΝ $X < 0$ ΤΟΤΕ

$Y \leftarrow A_T(X)$

ΑΛΛΙΩΣ_ΑΝ $X < 19$ ΤΟΤΕ

$Y \leftarrow 20 * X$

ΑΛΛΙΩΣ

$Y \leftarrow X - 1$

ΤΕΛΟΣ_ΑΝ

$B \leftarrow 1$

ΟΣΟ $B \leq Y$ ΕΠΑΝΑΛΑΒΕ

ΔΙΑΒΑΣΕ Z

$AB \leftarrow AB + Z$

$B \leftarrow B + 1$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ AB

ΜΕΧΡΙΣ_ΟΤΟΥ $AB \geq 2356$

Σε όλες τις παρακάτω ασκήσεις, ζητούμενο είναι η επαναδιατύπωση των εντολών με τέτοιο τρόπο ώστε να ακολουθούν τις αρχές του δομημένου προγραμματισμού (τι σημαίνει πρακτικά αυτό : εξαφανίστε την GOTO εξασφαλίζοντας ωστόσο το ίδιο τελικό αποτέλεσμα). Οι λύσεις δίνονται στο τέλος.

Άσκηση 17

```

μ ← 1
ΔΙΑΒΑΣΕ ν
5: AN μ > ν ΤΟΤΕ GOTO 7
ΓΡΑΨΕ μ+ν
μ ← μ+1
GOTO 5
7: ΓΡΑΨΕ ν^2

```

Λύση

```

μ ← 1
ΔΙΑΒΑΣΕ ν
ΟΣΟ μ <= ν ΕΠΑΝΑΛΑΒΕ
    ΓΡΑΨΕ μ+ν
    μ ← μ+1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ ν^2

```

Άσκηση 18

```

μ ← 100
4: ΔΙΑΒΑΣΕ ν
   AN ν MOD 7 = 0 ΤΟΤΕ GOTO 5
   μ ← μ+ν
   GOTO 6
5: μ ← μ-ν
6: AN μ <= 150 ΤΟΤΕ GOTO 4
   ΓΡΑΨΕ μ

```

Λύση

```

μ ← 100
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ ν
    AN ν MOD 7 = 0 ΤΟΤΕ
        μ ← μ-ν
    ΑΛΛΙΩΣ
        μ ← μ+ν
    ΤΕΛΟΣ_ΑΝ
ΜΕΧΡΙΣ_ΟΤΟΥ μ > 150
ΓΡΑΨΕ μ

```

ΆΛΥΤΕΣ ΑΣΚΗΣΕΙΣ ΣΤΗΝ GOTO

1. Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου που χρησιμοποιεί την goto σε τμήμα αλγορίθμου με δομημένο προγραμματισμό.

```

ΑΡΧΗ
Αν A>=0 τότε goto στο Βήμα 1
Βήμα 1: Αν A=0 τότε Εμφάνισε "Π1"
Βήμα 2: Εμφάνισε A
Πήγαινε στο Βήμα 4
Βήμα 3: Εμφάνισε -A
Βήμα 4: Εμφάνισε "Τέλος"
ΤΕΛΟΣ

```

2. Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου που χρησιμοποιεί την goto σε τμήμα αλγορίθμου με δομημένο προγραμματισμό.

```

K ← A
Βήμα 1: Εμφάνισε K
Βήμα 2: K ← K-B
Βήμα 3: Αν K<>1 τότε goto Βήμα 2
        Αλλιώς goto Βήμα 4
Βήμα 4: Εμφάνισε K

```

3. Να μετατραπεί το ακόλουθο τμήμα αλγορίθμου που χρησιμοποιεί την goto σε τμήμα αλγορίθμου με δομημένο προγραμματισμό.

```
Sum  $\leftarrow$  0  
i  $\leftarrow$  1  
Βήμα 1: Sum  $\leftarrow$  Sum +  $\Pi$  [i]  
i  $\leftarrow$  i+1  
Βήμα 2: Αν i  $\leq$  100 τότε  
    goto Βήμα 1  
Αλλιώς  
    goto Βήμα 3  
Τέλος_αν  
Βήμα 3: Εμφάνισε Sum
```

4. Θεωρήστε το ακόλουθο τμήμα εντολών:

```
20: Γράψε "Δώσε τιμή:"  
    Διάβασε x  
    Αν x=0 τότε goto 20
```

Να γραφεί ισοδύναμο τμήμα εντολών που να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

5. Θεωρήστε το ακόλουθο τμήμα εντολών:

```
K  $\leftarrow$  1  
S  $\leftarrow$  0  
20: Αν K > 100 τότε goto 40  
30: Διάβασε x  
    Αν x  $\leq$  0 τότε goto 30  
    S  $\leftarrow$  S+x  
    K  $\leftarrow$  K+1  
    goto 20  
40: Γράψε S
```

Να γραφεί ισοδύναμο τμήμα εντολών που να ακολουθεί τις αρχές του δομημένου προγραμματισμού.

- 1.** Ποιά στάδια περιλαμβάνει η επίλυση ενός προβλήματος με τον υπολογιστή;
- 2.** Ποιά είναι η εξέλιξη των γλωσσών προγραμματισμού;
- 3.** Τι είναι ένα πρόγραμμα σε γλώσσα μηχανής;
- 4.** Τι είναι οι συμβολικές γλώσσες και τι ο συμβολομεταφραστής;
- 5.** Ποιά τα μειονεκτήματα των συμβολικών γλωσσών;
- 6.** Αναφέρατε τρεις γλώσσες υψηλού επιπέδου.
- 7.** Για τι είναι κατάλληλη η FORTRAN;
- 8.** Σε τι προσανατολίζεται η COBOL;
- 9.** Η ALGOL είναι γλώσσα γενικής φύσης ή συνδέεται με συγκεκριμένες εφαρμογές;
- 10.** Η PL/1 είναι γλώσσα γενικής φύσης ή επιτρέπει παράλληλο προγραμματισμό;
- 11.** Η LISP είναι αντικειμενοστραφής γλώσσα ή προσανατολίζεται σε χειρισμό λιστών από σύμβολα;
- 12.** Η PROLOG προσανατολίζεται σε εμπορικές εφαρμογές ή σε προβλήματα τεχνητής νοημοσύνης;
- 13.** Για τι είναι κατάλληλη η BASIC; Τα προγράμματά της μεταγλωττίζονται ή εκτελούνται με τη βοήθεια διερμηνευτή;
- 14.** Για τι είναι κατάλληλη η PASCAL;
- 15.** Ποιό είναι το σημαντικότερο πλεονέκτημα της JAVA; Χρησιμοποιείται για αντικειμενοστραφή ή δομημένο προγραμματισμό;
- 16.** Η C είναι γλώσσα προγραμματισμού συστημάτων ή χειρισμού βάσεων δεδομένων;
- 17.** Τι σημαίνουν οι όροι προγραμματισμός οδηγούμενος από τα γεγονότα και οπτικός προγραμματισμός;
- 18.** Η dBASE είναι γλώσσα μαθηματικών υπολογισμών ή χειρισμού βάσεων δεδομένων;
- 19.** Ποια είναι τα πλεονεκτήματα της χρήσης γλωσσών υψηλού επιπέδου;
- 20.** Η SQL επιτρέπει οπτικό προγραμματισμό ή είναι γλώσσα χειρισμού βάσεων δεδομένων μέσω ερωταπαντήσεων;
- 21.** Τι είναι οπτικός προγραμματισμός και τι αντικειμενοστραφής;
- 22.** Ποιά είναι τα στοιχεία που προσδιορίζουν μία γλώσσα;
- 23.** Τι είναι το αλφάβητο μιας γλώσσας; Τι είναι το λεξιλόγιο μιας γλώσσας;
- 24.** Τι είναι το τυπικό και τι το συντακτικό της γραμματικής μιας γλώσσας; Τι σημαίνει σημασιολογία μιας γλώσσας;
- 25.** Ποιές είναι οι διαφορές φυσικών και τεχνητών γλωσσών;
- 26.** Είναι δυνατόν να εξελιχθεί μία γλώσσα προγραμματισμού;
- 27.** Ποιός είναι ο σκοπός της "από πάνω προς τα κάτω σχεδίασης προγραμμάτων";
- 28.** Ποιά τα πλεονεκτήματα του τμηματικού προγραμματισμού;
- 29.** Η εντολή GOTO που αλλάζει τη ροή εκτέλεσης ενός προγράμματος χρειάζεται στο δομημένο προγραμματισμό ή είναι περιττή;
- 30.** Δώστε τον ορισμό του δομημένου προγραμματισμού. Ποια είναι τα πλεονεκτήματά του;
- 31.** Ο αντικειμενοστραφής προγραμματισμός στηρίζει τη φιλοσοφία του στα δεδομένα ή στις ενέργειες που γίνονται πάνω σε αυτά;
- 32.** Μπορεί οποιοδήποτε πρόγραμμα να εκτελεστεί σε πολλούς επεξεργαστές ή πρέπει να έχει προγραμματιστεί με αυτό το σκοπό; ;
- 33.** Τι είναι ένας μεταγλωττιστής και τι είναι ένας διερμηνευτής;
- 34.** Περιγράψτε τα στάδια μεταγλώττισης και σύνδεσης ενός προγράμματος από το αρχικό πρόγραμμα ως το εκτελέσιμο
- 35.** Περιγράψτε τη διαδικασία μετάφρασης και εκτέλεσης ενός προγράμματος σε έναν διερμηνευτή

- 36.** Τι είναι ο συντάκτης και τι ο συνδέτης;
- 37.** Τι ονομάζεται πρόγραμμα;
- 38.** Τι είναι οι γλώσσες μηχανής;
- 39.** Ποιές οι διαφορές των γλωσσών υψηλού επιπέδου από αυτές χαμηλού επιπέδου;
- 40.** Ποιές γλώσσες υψηλού επιπέδου γνωρίζεις;
- 41.** Τι ονομάζουμε οπτικό προγραμματισμό και τι οδηγούμενο από τα γεγονότα;
- 42.** Πώς προσδιορίζεται μία φυσική γλώσσα;
- 43.** Ποιές οι κυριότερες διαφορές των φυσικών και των τεχνητών γλωσσών;
- 44.** Πώς γίνεται η παράσταση της ιεραρχικής σχεδίασης προγράμματος;
- 45.** Ποιές οι αρχές του δομημένου προγραμματισμού;
- 46.** Ποιά τα πλεονεκτήματα του δομημένου προγραμματισμού;
- 47.** Τι ονομάζεται αντικειμενοστραφής προγραμματισμός;
- 48.** Ποιά η διαδικασία για τη μετάφραση και εκτέλεση ενός προγράμματος;
- 49.** Ποιές οι διαφορές μεταγλωττιστή και διερμηνευτή;
- 50.** Ποιά προγράμματα και εργαλεία περιέχει ένα προγραμματιστικό περιβάλλον;

ΕΡΩΤΗΣΕΙΣ ΣΩΣΤΟΥ – ΛΑΘΟΥΣ

- 1.** Ο προγραμματισμός αφορά την διατύπωση ενός αλγορίθμου σε κατανοητή από τον υπολογιστή μορφή.
- 2.** Βασικό στοιχείο του προγράμματος είναι τα δεδομένα οι δομές δεδομένων πέραν της κωδικοποίησης.
- 3.** Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία ανθρώπου – μηχανής.
- 4.** Ένα πρόγραμμα γραμμένο σε γλώσσα μηχανής είναι εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες.
- 5.** Ένα πρόγραμμα σε γλώσσα μηχανής χρειάζεται μετατροπή σε ακολουθία δυαδικών ψηφίων ώστε να εκτελεστεί από τον υπολογιστή.
- 6.** Μια ακολουθία εντολών 0, 1 είναι κατανοητή από τον επεξεργαστή και μπορεί να εκτελεστεί χωρίς μετατροπή.
- 7.** Ο προγραμματισμός σε γλώσσα μηχανής ήταν μια εξαιρετικά δύσκολη δουλειά που ελάχιστοι μπορούσαν να πραγματοποιήσουν.
- 8.** Ένα πρόγραμμα σε συμβολική γλώσσα ή γλώσσα χαμηλού επιπέδου τελικά μετατρέπεται σε γλώσσα μηχανής.
- 9.** Η συμβολική γλώσσα είναι μια ακολουθία 0 και 1.
- 10.** Μια εντολή ενός προγράμματος γλώσσας χαμηλού επιπέδου μεταφράζεται σε γλώσσα μηχανής.
- 11.** Με τις γλώσσες χαμηλού επιπέδου το κέντρο βάρους μετακινήθηκε από τον υπολογιστή προς τον άνθρωπο.
- 12.** Οι εντολές σε συμβολική γλώσσα αποτελούνται από συμβολικά ονόματα που αντιστοιχούν σε εντολές της γλώσσας μηχανής.
- 13.** Οι συμβολικές γλώσσες έφεραν την ανεξαρτησία από την αρχιτεκτονική κάθε υπολογιστή.
- 14.** Τα προγράμματα σε γλώσσες χαμηλού επιπέδου είναι εύκολο να γραφτούν και να συντηρηθούν.
- 15.** Τα προγράμματα σε γλώσσες υψηλού επιπέδου είναι ανεξάρτητα του υπολογιστή που αναπτύχθηκαν.
- 16.** Οι γλώσσες υψηλού επιπέδου ήρθαν να επιλύσουν τις αδυναμίες των συμβολικών γλωσσών για καλύτερη επικοινωνία ανθρώπου – μηχανής.
- 17.** Η Fortran είναι γλώσσα χαμηλού επιπέδου.
- 18.** Η γλώσσα προγραμματισμού Fortran είναι κατάλληλη για την επίλυση όλων των ειδικών προβλημάτων.

19. Η Fortran είναι κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων.
20. Ένα πρόγραμμα γλώσσας υψηλού επιπέδου μπορεί να εκτελεστεί σε οποιονδήποτε υπολογιστή ανεξάρτητα της αρχιτεκτονικής.
21. Δεν έχει αναπτυχθεί γλώσσα υψηλού επιπέδου που να επιλύει όλα τα είδη προβλημάτων.
22. Η COBOL είναι γλώσσα προσανατολισμένη στην ανάπτυξη εμπορικών εφαρμογών.
23. Η COBOL δεν μπορεί να επιλύσει μαθηματικά προβλήματα.
24. Η Algol είναι μια γλώσσα γενικού σκοπού αλλά με ελάχιστη πρακτική εφαρμογή.
25. Η Algol αναπτύχθηκε για την επίλυση μαθηματικών προβλημάτων.
26. Η PL/1 είναι γλώσσα υψηλού επιπέδου γενικής χρήσης με ελάχιστη επιτυχία.
27. Η PL/1 προσπάθησε να συνθέσει Fortran και COBOL.
28. Η Lisp και η Prolog είναι γλώσσες προγραμματισμού που χρησιμοποιούνται στον τομέα της Τεχνητής Νοημοσύνης.
29. Η Basic είναι γλώσσα γενικού σκοπού με έμφαση στην εκπαίδευση αρχαρίων στον προγραμματισμό.
30. Η Basic κατόρθωσε με συνεχείς ανανεώσεις να καταστεί από τις πλέον δημοφιλείς γλώσσες προγραμματισμού.
31. Η γλώσσα Pascal είναι γλώσσα γενικού σκοπού και κατάλληλη τόσο για την εκπαίδευση όσο και για τη δημιουργία ισχυρών προγραμμάτων.
32. Η Pascal αποτέλεσε τη βάση για την παρουσίαση γλωσσών όπως η ADA και η Modula-2.
33. Η C χρησιμοποιήθηκε για την ανάπτυξη συστημάτων και έχει πολλές δυνατότητες χαμηλού επιπέδου.
34. Η C++ είναι γλώσσα τέταρτης γενιάς.
35. Η C++ και οι Java είναι αντικειμενοστραφείς γλώσσες.
36. Η Java είναι κατάλληλη για την ανάπτυξη εφαρμογών που εκτελούνται δικτυακά.
37. Ο οδηγούμενος από τα γεγονότα προγραμματισμός και οι γλώσσες οπτικού προγραμματισμού αξιοποιούν τα γραφικά χαρακτηριστικά του υπολογιστή (ποντίκι, μενού κ.λ.π.).
38. Στον οπτικό προγραμματισμό ο προγραμματιστής δεν γράφει κώδικα αλλά σχεδιάζει τα οπτικά αντικείμενα της εφαρμογής του.
39. Οι πιο διαδεδομένες οπτικές γλώσσες είναι η Visual Basic, η Visual C++ και η Java.
40. Ο δομημένος προγραμματισμός επιτρέπει την άμεση μεταφορά των αλγορίθμων σε πρόγραμμα.
41. Χάρη στο δομημένο προγραμματισμό δημιουργούνται προγράμματα απλούστερα, κατανοήσιμα και εύκολα στη διόρθωση.
42. Στο δομημένο προγραμματισμό ακολουθούνται οι αρχές του ιεραρχικού και του τμηματικού προγραμματισμού.
43. Στην ιεραρχική σχεδίαση, η ανάλυση του αλγορίθμου πραγματοποιείται με την τεχνική «από πάνω προς τα κάτω».
44. Ένα πλεονέκτημα των γλωσσών προγραμματισμού χαμηλού επιπέδου είναι η μεταφερσιμότητα των προγραμμάτων.
45. Παρά τη μεταφερσιμότητά τους, τα προγράμματα υψηλού επιπέδου είναι δυσκολότερο να διορθωθούν και να συντηρηθούν.
46. Στα πλεονεκτήματα των γλωσσών προγραμματισμού υψηλού επιπέδου σε σχέση με τις συμβολικές γλώσσες είναι η ανεξαρτησία από τον τύπο του υπολογιστή.
47. Οι γλώσσες υψηλού επιπέδου προτιμούνται για τον κοντύτερα στον άνθρωπο τρόπο έκφρασης;
48. Είναι ευκολότερη η εκμάθηση μιας γλώσσας χαμηλού επιπέδου
49. Ο παράλληλος προγραμματισμός προϋποθέτει την ύπαρξη περισσότερων από έναν επεξεργαστών.
50. Οι γλώσσες 4ης γενιάς χρησιμοποιούνται σε εφαρμογές που χρησιμοποιούν βάσεις δεδομένων.
51. Η καλύτερη γλώσσα προγραμματισμού είναι η Pascal.

52. Η επιλογή της καλύτερης γλώσσας προγραμματισμού εξαρτάται από το είδος της εφαρμογής.
53. Μια γλώσσα προγραμματισμού προσδιορίζεται από: το αλφάβητό της, το λεξιλόγιό της, τη γραμματική της και τη σημασιολογία της.
54. Λεξιλόγιο μιας γλώσσας είναι όλες οι ακολουθίες που δημιουργούνται από τα στοιχεία του αλφαβήτου της γλώσσας, τις λέξεις.
55. Η γραμματική είναι το συντακτικό μιας γλώσσας.
56. Δεν υπάρχουν διαφορές μεταξύ φυσικών και τεχνητών γλωσσών
57. Η ιεραρχική σχεδίαση διασπά ένα πρόβλημα σε υποπροβλήματα και τα επιλύει, λύνοντας το αρχικό πρόβλημα.
58. Ο τμηματικός προγραμματισμός υλοποιεί την φιλοσοφία της ιεραρχικής σχεδίασης.
59. Κάθε υποπρόβλημα της ιεραρχικής σχεδίασης αποτελεί μια ανεξάρτητη ενότητα στον τμηματικό προγραμματισμό.
60. Ο δομημένος προγραμματισμός περιέχει την ιεραρχική σχεδίαση και τον τμηματικό προγραμματισμό.
61. Ο δομημένος προγραμματισμός προϋποθέτει τη χρήση εξειδικευμένων αλγοριθμικών δομών.
62. Η αντικειμενοστραφής προσέγγιση εκλαμβάνει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τις τρεις αλγοριθμικές δομές: δομή ακολουθίας, δομή επιλογής και δομή επανάληψης.
63. Για την ανάπτυξη προγραμμάτων με τη φιλοσοφία του παράλληλου προγραμματισμού απαιτείται η χρήση εξειδικευμένων γλωσσών προγραμματισμού.
64. Ο διερμηνευτής (interpreter) μετατρέπει το εκτελέσιμο πρόγραμμα (executable) σε πρόγραμμα αντικείμενο (object).
65. Ο μεταγλωττιστής μας επιτρέπει να συντάσσουμε ένα πρόγραμμα.
66. Ο μεταγλωττιστής αναλαμβάνει το ρόλο της μετάφρασης από το πηγαίο πρόγραμμα στο εκτελέσιμο πρόγραμμα.
67. Ο διερμηνευτής ελέγχει και μετατρέπει μια-μια τις εντολές του πηγαίου προγράμματος σε γλώσσα μηχανής.
68. Ο συνδέτης μετατρέπει το πηγαίο πρόγραμμα σε βιβλιοθήκη.
69. Το αντικείμενο πρόγραμμα είναι ουσιαστικά γλώσσα μηχανής.
70. Ο συνδέτης είναι ένα πρόγραμμα ελέγχου των συντακτικών λαθών του πηγαίου προγράμματος.
71. Τα λογικά λάθη εντοπίζονται από έναν μεταγλωττιστή αλλά όχι από έναν διερμηνευτή.
72. Τα συντακτικά λάθη επιδιορθώνονται γρήγορα και εύκολα αλλά για τα λογικά λάθη η επιδιόρθωση είναι μια επίπονη διαδικασία.
73. Ο μεταγλωττιστής έχει το μειονέκτημα ότι ελέγχει όλο το πρόγραμμα και πραγματοποιεί και την διαδικασία της σύνδεσης πολλές φορές μέχρι να επιδιορθωθούν όλα τα λάθη.
74. Ο διερμηνευτής έχει το πλεονέκτημα ότι το πρόγραμμα εκτελείται γρηγορότερα.
75. Τα σύγχρονα προγραμματιστικά περιβάλλοντα χρησιμοποιούν μικτές υλοποιήσεις διερμηνευτή και μεταγλωττιστή.
76. Σ' ένα σύγχρονο προγραμματιστικό περιβάλλον απαιτείται η παρουσία συντάκτη για την συγγραφή του κειμένου των εντολών.
77. Για την επιδιόρθωση των λογικών λαθών πολλές φορές ο προγραμματιστής καλείται να εκτελέσει το πρόγραμμά του επανειλημμένα.
78. Ο μεταγλωττιστής σ' ένα σύγχρονο προγραμματιστικό περιβάλλον καθιστά την ύπαρξη του συνδέτη προαιρετική.
79. Ο μεταγλωττιστής διορθώνει όλα τα συντακτικά λάθη με τη χρήση βιβλιοθηκών.
80. Η παράλειψη μιας εντολής Τέλος_αν είναι λογικό λάθος.
81. Η χρήση της εντολής $MO \leftarrow \alpha + \beta + \gamma / 3$ αντί της $MO \leftarrow (\alpha + \beta + \gamma) / 3$ είναι λογικό λάθος.
82. Τα λογικά λάθη ενός προγράμματος εμφανίζονται κατά τη μεταγλώττιση.

- 83. Ο εντοπισμός των συντακτικών λαθών σε ένα πρόγραμμα γίνεται από τον μεταγλωττιστή (compiler).
- 84. Το εκτελέσιμο είναι το τελικό πρόγραμμα που εκτελείται από τον υπολογιστή.
- 85. Στις γλώσσες υψηλού επιπέδου δεν υπάρχει η ικανότητα της μεταφερισιμότητας.
- 86. Υπάρχουν δυο μεγάλες κατηγορίες μεταφραστικών προγραμμάτων: οι μεταφραστές και οι μεταγλωττιστές.
- 87. Ο μεταγλωττιστής εξάγει το πηγαίο πρόγραμμα.

ΕΡΩΤΗΣΕΙΣ ΣΗΜΠΛΗΡΩΣΗΣ ΚΕΝΩΝ

- 1. Το έργο της μετάφρασης εντολών γλώσσας χαμηλού επιπέδου σε ακολουθία 0, 1 αναλαμβάνει ο _____.
- 2. Η δυσκολία προγραμματισμού σε γλώσσα μηχανής οδήγησε στην δημιουργία _____ γλωσσών ή γλωσσών _____ επιπέδου.
- 3. Βασική τεχνική σχεδίασης προγραμμάτων είναι η τεχνική του _____ προγραμματισμού.
- 4. Κάθε πρόγραμμα γλώσσας υψηλού επιπέδου μεταφράζεται σε γλώσσα μηχανής από ένα ειδικό πρόγραμμα που ονομάζεται _____.
- 5. Η ανεξαρτησία των γλωσσών προγραμματισμού από την αρχιτεκτονική των υπολογιστών ονομάζεται _____.
- 6. Ο δομημένος προγραμματισμός στηρίζεται στη χρήση τριών στοιχειωδών δομών: της _____, της _____ και της _____.
- 7. _____ είναι το σύνολο των κανόνων που ορίζει τις μορφές που μια λέξη είναι αποδεκτή.
- 8. _____ είναι το σύνολο των κανόνων που καθορίζει τη διάταξη και τη σύνδεση των λέξεων.
- 9. Η λανθασμένη γραφή των δεσμευμένων λέξεων της γλώσσας προγραμματισμού είναι _____ λάθος.
- 10. _____ είναι το σύνολο των κανόνων που καθορίζει το νόημα των λέξεων και κατ' επέκταση των εκφράσεων και προτάσεων που χρησιμοποιούνται σε μια γλώσσα.
- 11. Κάθε _____ περιβάλλον διαθέτει τον μεταγλωττιστή του.
- 12. Ο μεταγλωττιστής δέχεται στην είσοδό του έναν πρόγραμμα σε γλώσσα _____ δημιουργώντας ισοδύναμο σε γλώσσα _____.
- 13. Η μεταγλώττιση ενός προγράμματος γίνεται από τους _____ ή τους _____.
- 14. Οι εντολές ενός προγράμματος μετατρέπονται σε ακολουθίες που αποτελούνται από 0 και 1, δηλαδή σε _____.
- 15. Τα _____ λάθη δεν είναι δυνατόν να εντοπίζονται από το προγραμματιστικό περιβάλλον.

1. Στον δομημένο προγραμματισμό χρησιμοποιούμε:
 - α) την εντολή GOTO
 - β) μόνο τις βασικές αλγοριθμικές δομές
 - γ) τις δομές δεδομένων
 - δ) τις βασικές αλγοριθμικές δομές και σπανίως την εντολή goto
2. Κατά την ανάλυση, ο κατακερματισμός ενός αλγορίθμου σε απλούστερους αποτελεί χαρακτηριστικό:
 - α) του τμηματικού προγραμματισμού
 - β) του ιεραρχικού προγραμματισμού
 - γ) της κλασσικής ανάπτυξης
 - δ) της υλοποίησης με την μέθοδο «από κάτω προς τα πάνω»
3. Χαρακτηριστικό του οπτικού προγραμματισμού είναι:
 - A. Επιτρέπει τη γραφική δημιουργία του περιβάλλοντος
 - B. Επιτρέπει την ανάπτυξη του προγράμματος σε τμήματα
 - Γ. Είναι ταχύτερος στην εκτέλεση των προγραμμάτων
 - Δ. Επιτρέπει την διαγραμματική παράσταση της σχεδίασης
4. Η Pascal είναι:
 - A. Κατάλληλη για εφαρμογές τεχνητής νοημοσύνης
 - B. Υποστηρίζει την ανάπτυξη παράλληλου προγραμματισμού
 - Γ. Μία γλώσσα γενικής χρήσης
 - Δ. Κατάλληλη μόνο για εκπαίδευση
5. Από τον συντάκτη παράγεται:
 - α) Τα αντικείμενα
 - β) ο εκτελέσιμος κώδικας
 - γ) ο πηγαίος κώδικας
 - δ) τίποτα από τα παραπάνω
6. Τα βήματα για τη δημιουργία ενός προγράμματος είναι:
 - α) συντάκτης
 - β) συνδέτης
 - γ) πηγαίο πρόγραμμα
 - δ) μεταγλωττιστής
 - ε) εκτελέσιμο πρόγραμμα
 - στ) αντικείμενο πρόγραμμα

ΕΡΩΤΗΣΕΙΣ ΑΝΤΙΣΤΟΙΧΗΣΗΣ

1. Να αντιστοιχίσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

Στήλη Α	Στήλη Β
1. Δομημένος προγραμματισμός 2. Τμηματικός προγραμματισμός	Α. Διάρθρωση του προβλήματος σε τμήματα που εκτελούνται παράλληλα Β. Ένα πρόβλημα περιγράφει "ενέργειες" που εφαρμόζονται στα δεδομένα Γ. Κάθε υποπρόβλημα αποτελεί ξεχωριστή ενότητα Δ. Χρήση βασικών αλγοριθμικών δομών

2. Να αντιστοιχίσετε τα στοιχεία της στήλης Α και της στήλης Β.

Α	Β
1. Δομή επιλογής	Α) Επαναληπτική εκτέλεση εντολών
2. Δομή επανάληψης	Β) Χρήση εντολή GoTo
3. Μη δομημένος προγραμματισμός	Γ) Σειριακή εκτέλεση εντολών
4. Δομή ακολουθίας	Δ) Επιλεκτική εκτέλεση εντολών
5. Δομημένος προγραμματισμός	Ε) Αποφυγή χρήσης Εντολής GoTo

3. Να αντιστοιχίσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

Στήλη Α	Στήλη Β
Α. Μεταγλωττιστής Β. Συνδέτης - Φορτωτής	1. Συμβολομεταφραστής 2. Αντικείμενο πρόγραμμα 3. Βιβλιοθήκες

4. Να αντιστοιχίσετε τα στοιχεία της στήλης Α με τα στοιχεία της στήλης Β.

Στήλη Α	Στήλη Β
Α. Διερμηνευτής Β. Μεταγλωττιστής Γ. Δομημένος προγραμματισμός Δ. Συντάκτης	1. Πρόγραμμα που δέχεται ένα πρόγραμμα γραμμένο σε γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής. 2. Μεθοδολογία σύνταξης προγραμμάτων που βοηθάει στην ανάπτυξη σύνθετων προγραμμάτων και στη διαχείρισή τους. 3. Πρόγραμμα που χρησιμοποιείται για τη σύνταξη και τη διόρθωση προγραμμάτων. 4. Πρόγραμμα που διαβάζει μια προς μια τις εντολές αρχικού προγράμματος και για κάθε μια εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.

ΚΕΦΑΛΑΙΟ 7

7.1 Το αλφάβητο της ΓΛΩΣΣΑΣ

Το αλφάβητο της **ΓΛΩΣΣΑΣ** αποτελείται από τα γράμματα του ελληνικού και του λατινικού αλφαβήτου, τα ψηφία, καθώς και από ειδικά σύμβολα, που χρησιμοποιούνται για προκαθορισμένες ενέργειες, στις οποίες θα αναφερθούμε στη συνέχεια.

7.2 Τύποι δεδομένων

Οι **τύποι δεδομένων** που υποστηρίζει η **ΓΛΩΣΣΑ** είναι:

- οι **αριθμητικοί**, που περιλαμβάνουν τους **ακέραιους** και τους **πραγματικούς** αριθμούς,
- οι **χαρακτήρες** [Οι χαρακτήρες πρέπει υποχρεωτικά να βρίσκονται μέσα σε απλά εισαγωγικά ' '. Τα δεδομένα αυτού του τύπου, επειδή περιέχουν τόσο αλφαβητικούς όσο και αριθμητικούς χαρακτήρες, ονομάζονται συχνά αλφαριθμητικά.] και
- οι **λογικοί**.

7.3 Σταθερές

Οι **σταθερές (constants)** είναι προκαθορισμένες τιμές που δεν μεταβάλλονται κατά τη διάρκεια εκτέλεσης του προγράμματος.

Οι **σταθερές** είναι τύπου δεδομένων, δηλαδή

- **ακέραιες,**
- **πραγματικές,**
- **αλφαριθμητικές ή χαρακτήρες**
- **λογικές.**

Συμβολικές σταθερές

Η **ΓΛΩΣΣΑ** επιτρέπει την αντιστοίχιση σταθερών τιμών με ονόματα, εφόσον αυτά δηλωθούν στην αρχή του προγράμματος (στο τμήμα δήλωσης σταθερών).

ΣΤΑΘΕΡΕΣ

$\pi=3.14159$

$\Phi\pi\alpha=0.19$

$\text{ΟΝΟΜΑ}=\text{'Κώστας'}$

Η χρήση σταθερών κάνει το πρόγραμμα πιο κατανοητό και κατά συνέπεια ευκολότερο να διορθωθεί και να συντηρηθεί.

7.4 Μεταβλητές

Η έννοια της μεταβλητής (**variable**) είναι γνωστή από τα μαθηματικά. Για παράδειγμα ο τύπος της γεωμετρίας $E = \alpha\beta$ υπολογίζει το εμβαδόν (**E**) ενός ορθογωνίου με διαστάσεις, που συμβολίζονται με α και β . Αν στο α και στο β δοθούν οι αντίστοιχες τιμές, τότε ο τύπος αυτός υπολογίζει το εμβαδόν του ορθογωνίου.

Μια μεταβλητή λοιπόν, παριστάνει μία ποσότητα που η τιμή της μπορεί να μεταβάλλεται αυτό που μένει υποχρεωτικά αναλλοίωτο είναι ο τύπος της μεταβλητής.

Η ΓΛΩΣΣΑ επιτρέπει τη χρήση μεταβλητών των τεσσάρων τύπων που αναφέρθηκαν, δηλαδή ακεραίων, πραγματικών, χαρακτήρων και λογικών, ενώ η δήλωση του τύπου κάθε μεταβλητής γίνεται υποχρεωτικά στο τμήμα δήλωσης μεταβλητών.

Το όνομα κάθε μεταβλητής ακολουθεί τους κανόνες δημιουργίας ονομάτων, δηλαδή αποτελείται από γράμματα, ψηφία καθώς και τον χαρακτήρα `_`, ενώ το όνομα κάθε μεταβλητής είναι μοναδικό για κάθε πρόγραμμα.

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Όνομα

ΛΟΓΙΚΕΣ: Έλεγχος

ΑΚΕΡΑΙΕΣ: ΤΙΜΗ, N

ΠΡΑΓΜΑΤΙΚΕΣ: Εμβαδόν, A

Αν και όπως αναφέρθηκε, το όνομα των μεταβλητών μπορεί να είναι οποιοσδήποτε συνδυασμός χαρακτήρων, είναι καλή πρακτική να χρησιμοποιούνται ονόματα τα οποία να υπονοούν το περιεχόμενό τους, κάνοντας το πρόγραμμα ευκολότερο στην ανάγνωσή του και στην κατανόησή του.

7.5 Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές που υποστηρίζονται από τη ΓΛΩΣΣΑ καλύπτουν τις βασικές πράξεις: πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση, ενώ υποστηρίζεται και η ύψωση σε δύναμη, η ακέραια διαίρεση και το υπόλοιπο της ακέραιας διαίρεσης.

Αριθμητικός τελεστής	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη
DIV	Ακέραια διαίρεση
MOD	Υπόλοιπο ακέραιας διαίρεσης

7.6 Συναρτήσεις

Πολλές γνωστές συναρτήσεις από τα μαθηματικά χρησιμοποιούνται συχνά και περιέχονται στη ΓΛΩΣΣΑ. Οι συναρτήσεις αυτές είναι:

HM(X)	Υπολογισμός ημιτόνου
ΣΥΝ(X)	Υπολογισμός συνημιτόνου
ΕΦ(X)	Υπολογισμός εφαπτομένης
T_P(X)	Υπολογισμός τετραγωνικής ρίζας
ΛΟΓ(X)	Υπολογισμός φυσικού λογαρίθμου
E(X)	Υπολογισμός του ex
A_M(X)	Ακέραιο μέρος του X
A_T(X)	Απόλυτη τιμή του X

7.7 Αριθμητικές εκφράσεις

Όταν μια τιμή προκύπτει από υπολογισμό, τότε αναφερόμαστε σε εκφράσεις (**expressions**). Για τη σύνταξη μιας αριθμητικής έκφρασης χρησιμοποιούνται αριθμητικές σταθερές, μεταβλητές, συναρτήσεις, αριθμητικοί τελεστές και παρενθέσεις. Οι αριθμητικές εκφράσεις υλοποιούν απλές ή σύνθετες μαθηματικές πράξεις.

Κάθε έκφραση παριστάνει μια συγκεκριμένη αριθμητική τιμή, η οποία βρίσκεται μετά την εκτέλεση των πράξεων. Γι' αυτό είναι απαραίτητο όλες οι μεταβλητές, που εμφανίζονται σε μια έκφραση να έχουν οριστεί προηγούμενα, δηλαδή να έχουν κάποια τιμή.

Ιεραρχία

1. Ύψωση σε δύναμη.
2. Πολλαπλασιασμός και διαίρεση.
3. Πρόσθεση και αφαίρεση.

Όταν η ιεραρχία είναι ίδια, τότε οι πράξεις εκτελούνται από τ' αριστερά προς τα δεξιά. Σε πολλές όμως περιπτώσεις είναι απαραίτητο να προηγηθεί μια πράξη χαμηλότερης ιεραρχίας. Αυτό επιτυγχάνεται με την εισαγωγή των παρενθέσεων. Η πράξη που πρέπει να προηγηθεί περικλείεται σε ένα ζεύγος παρενθέσεων, οπότε και εκτελείται πρώτη.

Π.χ. η έκφραση $2 + 3 * 4$ δίδει ως αποτέλεσμα 14, ενώ η $(2 + 3) * 4$ δίδει 20, διότι εκτελείται πρώτα η πρόσθεση και μετά ο πολλαπλασιασμός.

7.8 Εντολή εκχώρησης

Η εντολή εκχώρησης χρησιμοποιείται για την απόδοση τιμών στις μεταβλητές κατά τη διάρκεια εκτέλεσης του προγράμματος.

7.9 Εντολές εισόδου εξόδου

Η ΓΛΩΣΣΑ υποστηρίζει για την εισαγωγή δεδομένων από το πληκτρολόγιο την εντολή **ΔΙΑΒΑΣΕ** και για την εμφάνιση των αποτελεσμάτων την εντολή **ΓΡΑΨΕ**.

Η εντολή **ΔΙΑΒΑΣΕ** ακολουθείται πάντοτε από ένα ή περισσότερα ονόματα μεταβλητών. Αν υπάρχουν περισσότερες από μία μεταβλητές τότε αυτές χωρίζονται με **κόμμα(,)**.

Κατά την εκτέλεση του προγράμματος η εντολή **ΔΙΑΒΑΣΕ** διακόπτει την εκτέλεσή του και το πρόγραμμα περιμένει την εισαγωγή από το πληκτρολόγιο τιμών, που θα εκχωρηθούν στις μεταβλητές. Μετά την ολοκλήρωση της εντολής η εκτέλεση του προγράμματος συνεχίζεται με την επόμενη εντολή.

Παράδειγμα: ΔΙΑΒΑΣΕ Ποσότητα, Τιμή

Η εντολή **ΓΡΑΨΕ** έχει ως αποτέλεσμα την εμφάνιση τιμών στη μονάδα εξόδου. Συσκευή εξόδου μπορεί να είναι η οθόνη του υπολογιστή, ο εκτυπωτής, βοηθητική μνήμη ή γενικά οποιαδήποτε συσκευή εξόδου έχει οριστεί στο πρόγραμμα.

Για τα παραδείγματα αυτού του κεφαλαίου θεωρούμε ότι η εμφάνιση γίνεται πάντοτε στην οθόνη. Η λίστα των στοιχείων μπορεί να περιέχει σταθερές τιμές και ονόματα μεταβλητών.

Κατά την εκτέλεση του προγράμματος η εντολή **ΓΡΑΨΕ** προκαλεί την εμφάνιση στην οθόνη των σταθερών τιμών. Όταν κάποιο όνομα μεταβλητής περιέχεται στη λίστα, τότε αρχικά ανακτάται η τιμή της και στη συνέχεια η τιμή αυτή εμφανίζεται στην οθόνη.

Η χρήση της εντολής **ΓΡΑΨΕ** είναι κυρίως η εμφάνιση μηνυμάτων από τον υπολογιστή, καθώς και αποτελεσμάτων που περιέχονται στις μεταβλητές.

7.10 Δομή προγράμματος

Η πρώτη εντολή κάθε προγράμματος είναι υποχρεωτικά η επικεφαλίδα του προγράμματος, η οποία είναι η λέξη **ΠΡΟΓΡΑΜΜΑ** ακολουθούμενη από το όνομα του προγράμματος. Το τελευταίο πρέπει να υπακούει στους κανόνες δημιουργίας ονομάτων της **ΓΛΩΣΣΑΣ**.

Στη συνέχεια ακολουθεί το τμήμα δήλωσης των **σταθερών** του προγράμματος, αν βέβαια το πρόγραμμα μας χρησιμοποιεί σταθερές.

Αμέσως μετά είναι το τμήμα δήλωσης μεταβλητών, όπου δηλώνονται υποχρεωτικά τα ονόματα όλων των **μεταβλητών** καθώς και ο τύπος τους.

Ακολουθεί το κύριο μέρος του προγράμματος, που περιλαμβάνει όλες τις εκτελέσιμες εντολές. Οι εντολές αυτές περιλαμβάνονται υποχρεωτικά ανάμεσα στις λέξεις **ΑΡΧΗ** και **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ**.

Τέλος αν το πρόγραμμα χρησιμοποιεί διαδικασίες ή συναρτήσεις (**βλέπε κεφ. 10**), αυτές γράφονται μετά το **ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ**.

Κάθε εντολή γράφεται σε ξεχωριστή γραμμή. Αν μία εντολή πρέπει να συνεχιστεί και στην επόμενη γραμμή, τότε ο πρώτος χαρακτήρας αυτής της γραμμής πρέπει να είναι ο χαρακτήρας **&**.

Αν ο πρώτος χαρακτήρας είναι το θαυμαστικό (**!**), σημαίνει ότι αυτή η γραμμή περιέχει σχόλια και όχι εκτελέσιμες εντολές.

ΚΕΦΑΛΑΙΟ 8

8.1 Εντολές Επιλογής

A. Λογική Έκφραση

Για τη σύνταξη μιας λογικής έκφρασης ή συνθήκης χρησιμοποιούνται **σταθερές, μεταβλητές, αριθμητικές** παραστάσεις, **συγκριτικοί** και **λογικοί** τελεστές, καθώς και παρενθέσεις. Στις λογικές εκφράσεις γίνεται σύγκριση της τιμής μίας έκφρασης, που βρίσκεται αριστερά από το συγκριτικό τελεστή με την τιμή μιας άλλης έκφρασης που βρίσκεται δεξιά. Το αποτέλεσμα είναι μία λογική τιμή **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**.

Συγκριτικοί τελεστές		
Τελεστής	Ελεγχόμενη σχέση	Παράδειγμα
=	Ισότητα	Αριθμός=0
<>	Ανισότητα	Όνομα1 <> 'Κώστας'
>	Μεγαλύτερο από	Τιμή>10000
>=	Μεγαλύτερο ή ίσο	$X+Y \geq (A+B)/\Gamma$
<	Μικρότερο από	$B^2-4*A*\Gamma < 0$
<=	Μικρότερο ή ίσο	Βάρος <= 500

Οι συγκρίσεις γίνονται σε δεδομένα αριθμητικά, αλφαριθμητικά και λογικά.

Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μια έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες. Ακόμη, οι λογικοί τελεστές έχουν χαμηλότερη ιεραρχία από τους συγκριτικούς.

Η σύγκριση μεταξύ δύο αριθμών γίνεται με προφανή τρόπο. Στην περίπτωση των πραγματικών αριθμών θεωρούμε ότι οι αριθμοί μπορούν να έχουν άπειρο αριθμό ψηφίων.

Η σύγκριση ατομικών χαρακτήρων στηρίζεται στην αλφαβητική σειρά, για παράδειγμα το 'α' θεωρείται μικρότερο από το 'β'.

Η σύγκριση αλφαριθμητικών δεδομένων βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση μέχρις ότου βρεθεί κάποια διαφορά, για παράδειγμα η λέξη 'κακός' θεωρείται μικρότερη από τη λέξη 'καλός' αφού το γράμμα κ προηγείται του γράμματος λ.

Η σύγκριση λογικών έχει έννοια μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>), αφού οι τιμές που μπορούν να έχουν είναι **ΑΛΗΘΗΣ** και **ΨΕΥΔΗΣ**.

B. Σύνθετες Εκφράσεις

Σε πολλά προβλήματα οι επιλογές δεν αρκεί να γίνονται με απλές **λογικές** παραστάσεις όπως αυτές οι οποίες αναφέρθηκαν, αλλά χρειάζεται να συνδυαστούν μία ή περισσότερες **λογικές** παραστάσεις. Αυτό επιτυγχάνεται με τη χρήση των **τριών** βασικών λογικών τελεστών **ΟΧΙ**, **ΚΑΙ**, **Ή**.

Παραδείγματα

$0 < X < 5$

$X = 1$ ή 2 ή 3

$X > 0$ **ΚΑΙ** $X < 5$

$X = 1$ **Ή** $X = 2$ **Ή** $X = 3$

8.1.1 Εντολή **ΑΝ**

Η δομή επιλογής υλοποιείται στη **ΓΛΩΣΣΑ** με την εντολή **ΑΝ**.

Συχνά η εντολή **ΑΝ** εκτός από το τμήμα των εντολών, που εκτελούνται όταν η λογική έκφραση είναι Αληθής, περιέχει και το τμήμα των εντολών που εκτελούνται, αν δεν ισχύει η συνθήκη (είναι Ψευδής).

Η εντολή **ΑΝ** εμφανίζεται με τρεις διαφορετικές μορφές.

Α) Την απλή εντολή **ΑΝ...ΤΟΤΕ**,

Β) Την σύνθετη εντολή **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ** και τέλος

Γ) Την πολλαπλή εντολή **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ_ΑΝ**.

και **ΕΠΙΛΕΞΕ** έκφραση **ΠΕΡΙΠΤΩΣΗ** λίστα_τιμ1...**ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ** εντολές_αλλιώς

Κάθε εντολή **ΑΝ** πρέπει να κλείνει με **ΤΕΛΟΣ_ΑΝ**.

Α) i. Λειτουργία

Αν η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΤΕΛΟΣ_ΑΝ**, σε αντίθετη περίπτωση οι εντολές αυτές αγνοούνται. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ_ΑΝ**

ii. Σύνταξη

ΑΝ συνθήκη ΤΟΤΕ

εντολή-1

εντολή-2

...

εντολή-n

ΤΕΛΟΣ_ΑΝ

Β) i. Λειτουργία

Αν η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΑΛΛΙΩΣ**, διαφορετικά εκτελούνται οι εντολές μεταξύ **ΑΛΛΙΩΣ** και **ΤΕΛΟΣ_ΑΝ**. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ_ΑΝ**

ii. Σύνταξη

ΑΝ συνθήκη ΤΟΤΕ

εντολή-1

εντολή-2

...

εντολή-n

ΑΛΛΙΩΣ

εντολή-1

εντολή-2

...

εντολή-ν

ΤΕΛΟΣ_ΑΝ
Γ) i. Λειτουργία

Εκτελούνται οι εντολές που βρίσκονται στο αντίστοιχο τμήμα, όταν η συνθήκη είναι αληθής. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ_ΑΝ**

ii. Σύνταξη ΑΝ συνθήκη-1 ΤΟΤΕ εντολή-1 ... εντολή-ν ΑΛΛΙΩΣ_ΑΝ συνθήκη-2 ΤΟΤΕ εντολή-1 ... εντολή-ν ΑΛΛΙΩΣ εντολή-1 ... εντολή-ν ΤΕΛΟΣ_ΑΝ	Η Εντολή ΕΠΙΛΕΞΕ Σύνταξη ΔΙΑΒΑΣΕ ΜΕΤΑΒΛΗΤΗ ΕΠΙΛΕΞΕ ΜΕΤΑΒΛΗΤΗ ΠΕΡΙΠΤΩΣΗ ΣΥΝΘΗΚΗ1 Η ΤΙΜΗ Η ΑΠΟ..ΕΩΣ ΕΝΤΟΛΗ ΠΕΡΙΠΤΩΣΗ ΣΥΝΘΗΚΗ2 Η ΤΙΜΗ Η ΑΠΟ..ΕΩΣ ΕΝΤΟΛΗ ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ ΕΝΤΟΛΗ ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
--	---

Υπολογίζεται η τιμή της έκφρασης και εκτελούνται οι εντολές που ανήκουν στην αντίστοιχη περίπτωση τιμών. Αν η τιμή της έκφρασης δεν αντιστοιχεί σε καμία περίπτωση, τότε εκτελούνται οι εντολές αλλιώς.

Αν οι εναλλακτικές περιπτώσεις επιλογής είναι πολλές, μπορεί να χρησιμοποιηθεί η εντολή **ΕΠΙΛΕΞΕ**, η γενική μορφή της οποίας είναι:

Να αποφεύγεται, αν είναι δυνατόν, η χρήση των εμφωλευμένων **ΑΝ**, και στη θέση τους να χρησιμοποιούνται απλούστερες δομές που διευκολύνουν την ανάγνωση και την κατανόηση του προγράμματος.

Στην εντολή αυτή οι λίστες τιμών που συνοδεύουν κάθε περίπτωση μπορούν να περιλαμβάνουν μία ή περισσότερες τιμές ή περιοχή τιμών από-έως.

Η χρήση της εντολής **ΕΠΙΛΕΞΕ** λόγω της συμπαγούς δομής της προσφέρει σημαντικά πλεονεκτήματα στον προγραμματισμό.

8.2 Εντολές επανάληψης

Η **ΓΛΩΣΣΑ** υποστηρίζει τρεις εντολές επανάληψης, την εντολή **ΟΣΟ** όπου η επανάληψη ελέγχεται από μία λογική έκφραση στην αρχή και εκτελείται συνεχώς όσο η συνθήκη είναι Αληθής, την εντολή **ΜΕΧΡΙΣ_ΟΤΟΥ** όπου η συνθήκη βρίσκεται στο τέλος του βρόχου και εκτελείται συνεχώς μέχρις ότου η συνθήκη αυτή γίνει Αληθής και τέλος την εντολή **ΓΙΑ**, με την οποία ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό φορών.

8.2.1 Εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

i. Λειτουργία

Ελέγχεται η συνθήκη και αν είναι Αληθής, εκτελούνται οι εντολές που βρίσκονται ανάμεσα στις **ΟΣΟ_ΕΠΑΝΑΛΑΒΕ** και **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**.

Στη συνέχεια ελέγχεται πάλι η συνθήκη και αν ισχύει, εκτελούνται πάλι οι ίδιες εντολές. Όταν η λογική έκφραση γίνει Ψευδής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**.

ii. Σύνταξη

ΟΣΟ συνθήκη **ΕΠΑΝΑΛΑΒΕ**

εντολή-1

εντολή-2

...

εντολή-n

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Εφόσον μετά από κάθε επανάληψη ελέγχεται εκ νέου η συνθήκη, πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μία εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Σε αντίθετη περίπτωση η επανάληψη δε θα τερματίζεται και θα εκτελείται συνεχώς.

Η χρήση τιμών για τον τερματισμό μίας επαναληπτικής διαδικασίας, όπως στο παράδειγμα η αυθαίρετη επιλογή του 0, είναι συνήθης στον προγραμματισμό.

Η τιμή αυτή ορίζεται από τον προγραμματιστή και αποτελεί μια σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή είναι τέτοια, ώστε να μην είναι λογικά σωστή για το πρόβλημα, για παράδειγμα η τιμή 0 αποκλείεται από τις μετρήσεις σύμφωνα με την εκφώνηση του παραδείγματος.

Η τιμή αυτή συχνά αποκαλείται "τιμή φρουρός".

8.2.2 Εντολή ΜΕΧΡΙΣ_ΟΤΟΥ

i. Λειτουργία

Εκτελούνται οι εντολές μεταξύ των **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**. Στη συνέχεια ελέγχεται η λογική έκφραση και αν δεν ισχύει (είναι ψευδής), τότε οι εντολές που βρίσκονται ανάμεσα στις **ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**, εκτελούνται πάλι. Ελέγχεται ξανά η λογική έκφραση και αν δεν ισχύει, επαναλαμβάνεται η εκτέλεση των ίδιων εντολών.

Όταν η λογική έκφραση γίνει Αληθής τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά από την **ΜΕΧΡΙΣ_ΟΤΟΥ**.

Εκτελούνται οι εντολές μεταξύ των ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ και ΜΕΧΡΙΣ_ΟΤΟΥ. Στη συνέχεια ελέγχεται η λογική έκφραση και αν δεν ισχύει (είναι ψευδής), τότε οι εντολές που βρίσκονται ανάμεσα στις ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ και ΜΕΧΡΙΣ_ΟΤΟΥ, εκτελούνται πάλι.

Ελέγχεται ξανά η λογική έκφραση και αν δεν ισχύει, επαναλαμβάνεται η εκτέλεση των ίδιων εντολών. Όταν η λογική έκφραση γίνει Αληθής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά από την ΜΕΧΡΙΣ_ΟΤΟΥ.

Πολύ συχνά η ίδια επαναληπτική διαδικασία μπορεί να γραφεί εξίσου σωστά χρησιμοποιώντας είτε τη δομή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ είτε τη δομή ΜΕΧΡΙΣ_ΟΤΟΥ και είναι προσωπική επιλογή του προγραμματιστή ποια από τις δυο θα χρησιμοποιήσει.

Υπάρχουν όμως περιπτώσεις όπου η χρήση της εντολής ΜΕΧΡΙΣ_ΟΤΟΥ οδηγεί σε απλούστερα και πιο ευκολονόητα προγράμματα. Γενικά σε περιπτώσεις όπου η επανάληψη θα συμβεί υποχρεωτικά μία φορά, είναι προτιμότερη η χρήση της ΜΕΧΡΙΣ_ΟΤΟΥ.

Χαρακτηριστική περίπτωση όπου προτιμάται η εντολή ΜΕΧΡΙΣ_ΟΤΟΥ είναι στον έλεγχο αποδεκτών τιμών καθώς και στην επιλογή από προκαθορισμένες απαντήσεις ή μενού

ii. Σύνταξη

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

εντολή-1

εντολή-2

...

εντολή-n

ΜΕΧΡΙΣ_ΟΤΟΥ λογική-έκφραση

8.2.3 Εντολή ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ

i. Λειτουργία

Οι εντολές του βρόχου εκτελούνται για όλες τις τιμές της μεταβλητής από την αρχική τιμή μέχρι την τελική τιμή, αυξανόμενες με την τιμή του βήματος. Αν το βήμα είναι ίσο με 1, τότε παραλείπεται.

ii. Σύνταξη

ΓΙΑ μεταβλητή **ΑΠΟ** τιμή1 **ΜΕΧΡΙ** τιμή2 **ΜΕ ΒΗΜΑ** τιμή3

εντολή-1

εντολή-2

...

εντολή-n

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Πολύ συχνά για την επίλυση των προβλημάτων απαιτείται η χρήση **εμφωλευμένων βρόχων**. Σε αυτή την περίπτωση ο ένας βρόχος βρίσκεται μέσα στον άλλο.

Στη χρήση των **εμφωλευμένων βρόχων** ισχύουν συγκεκριμένοι κανόνες που πρέπει να ακολουθούνται αυστηρά για την σωστή λειτουργία των προγραμμάτων.

Συγκεκριμένα:

- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος, πρέπει να ολοκληρώνεται πρώτος.
- Η είσοδος σε κάθε βρόχο υποχρεωτικά γίνεται από την αρχή του.
- Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

ΓΕΝΙΚΕΣ ΕΡΩΤΗΣΕΙΣ ΣΤΑ ΚΕΦΑΛΑΙΑ 7 - 8

88. Οι τύποι μεταβλητών που δέχεται η ΓΛΩΣΣΑ είναι μόνο ΠΡΑΓΜΑΤΙΚΕΣ και ΑΚΕΡΑΙΕΣ.
89. Οι δηλώσεις των σταθερών προηγούνται πάντοτε των δηλώσεων των μεταβλητών.
90. Τα σχόλια τοποθετούνται πάντα στην αρχή του προγράμματος.
91. Κάθε εντολή ΑΝ περιλαμβάνει υποχρεωτικά το τμήμα ΑΛΛΙΩΣ.
92. Κάθε τμήμα προγράμματος που χρησιμοποιεί την εντολή ΕΠΙΛΕΞΕ μπορεί να γραφεί και με εντολές ΑΝ.
93. Η χρήση εμφωλευμένων ΑΝ είναι καλή προγραμματιστική τακτική.
94. Αν το Α έχει την τιμή 10 και το Β την τιμή 20 τότε η έκφραση $(A > 8 \text{ ΚΑΙ } B < 20) \text{ Ή } (A > 10 \text{ Ή } B = 10)$ είναι αληθής.
95. Οι εντολές που βρίσκονται σε μία επανάληψη ΓΙΑ εκτελούνται τουλάχιστον μία φορά.
96. Κάθε επανάληψη μπορεί να γραφεί με την εντολή ΟΣΟ – ΕΠΑΝΑΛΑΒΕ.
97. Σε περίπτωση εμφωλευμένων βρόχων, ο εσωτερικός πρέπει να περικλείεται ολόκληρος στον εξωτερικό.
98. Η τιμή του βήματος αναφέρεται υποχρεωτικά σε κάθε εντολή ΓΙΑ.
99. Οι μεταβλητές που χρησιμοποιούνται σε ένα πρόγραμμα αντιστοιχούνται από το μεταγλωττιστή σε συγκεκριμένες θέσεις μνήμης του Η/Υ.
100. Η ταξινόμηση των στοιχείων ενός πίνακα πρέπει να γίνεται πάντα πριν από την αναζήτηση.
101. Δεσμευμένες λέξεις καλούνται οι λέξεις που έχουν δεσμεύσει για τα ονόματα των μεταβλητών.
102. Το αλφάβητο της γλώσσας αποτελείται μόνο από γράμματα ελληνικά - λατινικά και αριθμούς.
103. Σε μια εντολή εκχώρησης η μεταβλητή αριστερά και η έκφραση δεξιά του βέλους πρέπει να είναι του ίδιου τύπου.
104. Στη δομή ενός προγράμματος σε ΓΛΩΣΣΑ είναι αυστηρά καθορισμένη.
105. Το τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ πρέπει απαραίτητως να προηγείται του τμήματος εντολών.
106. Μια έκφραση μπορεί να περιέχει μεταβλητές, σταθερές, τελεστές και παρενθέσεις.
107. Όταν μια δομή επανάληψης είναι εμφωλευμένη σε μια άλλη, τότε για κάθε εξωτερικό βρόχο πρέπει να ολοκληρώνονται όλες οι επαναλήψεις του εσωτερικού.
108. Όταν μια δομή "Για" είναι εμφωλευμένη σε μια άλλη δομή "Για", τότε μπορούμε αν το επιθυμούμε για ευκολία να χρησιμοποιήσουμε την ίδια μεταβλητή για μετρητή.
109. Σε οποιοδήποτε σημείο ενός προγράμματος σε ΓΛΩΣΣΑ μπορούν να τοποθετηθούν σχόλια.
110. Μπορούμε να αρχικοποιούμε μεταβλητές στο τμήμα δηλώσεων ενός προγράμματος σε ΓΛΩΣΣΑ
111. Σε ένα πρόγραμμα σε ΓΛΩΣΣΑ δεν μπορούμε να χρησιμοποιήσουμε λογική σταθερά.
112. Ποιά από τις παρακάτω εντολές αυξάνει τη μεταβλητή Πλήθος κατά μία μονάδα

A) Πλήθος $\leftarrow$ Πλήθος+1	Γ) Πλήθος $\leftarrow$ 1
B) Πλήθος $\leftarrow$ +1	Δ) Πλήθος+1 $\leftarrow$ Πλήθος

113. Ποιά η τιμή της μεταβλητής Α μετά την εκτέλεση της παρακάτω εντολής:

$$\text{Τιμή} \leftarrow (5 + 4 / 2 * 2) * 2 - (3 * 2 + 5 - 3)^2 + 9 / 3 - 2.$$

A) -53	Γ) -125
B) -37	Δ) -45

114. Να συμπληρωθούν τα κενά:

ΠΡΟΓΡΑΜΜΑ ΤΕΣΤ

Π = 3.14

ΜΕΤΑΒΛΗΤΕΣ

_____ : Ε, Ακτίνα

ΑΡΧΗ

ΔΙΑΒΑΣΕ Ακτίνα

Ε $\leftarrow$ Π * Ακτίνα ²

ΓΡΑΨΕ 'Εμβαδόν :', _____

115. Τι θα εκτυπώσουν οι παρακάτω εντολές:

A $\leftarrow$ 0

B $\leftarrow$ 5

Γ $\leftarrow$ 10

ΑΝ A > 10 **ΤΟΤΕ**

ΑΝ B > 20 **ΤΟΤΕ**

ΑΝ Γ > 10 **ΤΟΤΕ**

ΓΡΑΨΕ Γ

ΑΛΛΙΩΣ

ΓΡΑΨΕ 2 * Γ

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

ΓΡΑΨΕ B

ΤΕΛΟΣ_ΑΝ

ΑΛΛΙΩΣ

ΑΝ B < 10 **ΤΟΤΕ**

ΓΡΑΨΕ A

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

A. 0	Γ. 5
B. 10	Δ. 20

116. Να συμπληρωθούν τα κενά ώστε οι επόμενες εντολές να τυπώνουν πάντα τον μεγαλύτερο αριθμό από τους δύο που διαβάστηκαν.

ΔΙΑΒΑΣΕ A,B

ΑΝ A < B _____

ΤΕΛΟΣ_ΑΝ

ΓΡΑΨΕ A

117. Να συμπληρωθούν τα κενά ώστε οι επόμενες εντολές να τυπώνουν την τετραγωνική ρίζα.

```

ΔΙΑΒΑΣΕ Α
ΑΝ Α < 0 ΤΟΤΕ
    Ρίζα ← T_P(A)
    ΓΡΑΨΕ Ρίζα
    _____
    ΓΡΑΨΕ 'Δεν υπάρχει ρίζα'
ΤΕΛΟΣ_ΑΝ
    
```

118. Πόσες φορές θα εκτελεστεί η εντολή ΓΡΑΨΕ Α.

```

Α ← 10
ΟΣΟ Α <> 0 ΕΠΑΝΑΛΑΒΕ
    ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5
        Α ← Α-1
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ Α
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    
```

A. 10	Γ. 2
B. 0	Δ. Άπειρες

119. Να συμπληρωθούν τα κενά ώστε οι επόμενες εντολές να τυπώνουν το άθροισμα των τετραγώνων των περιττών αριθμών που είναι μικρότεροι από 10.

```

Άθροισμα <- _____
ΓΙΑ _____ ΑΠΟ 1 ΜΕΧΡΙ 10 ΜΕ_ΒΗΜΑ _____
    Άθροισμα ← _____ + Ι ^ 2
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ Άθροισμα
    
```

120. Να συμπληρωθούν τα κενά ώστε οι επόμενες εντολές να τυπώνουν το άθροισμα των αριθμών από 100 έως 200.

```

Κ ← _____
Σ ← _____
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    Σ ← Σ+Κ
    Κ ← Κ+1
ΜΕΧΡΙΣ_ΟΤΟΥ _____
ΓΡΑΨΕ Σ
    
```

121. Ποιές από τις παρακάτω εντολές υπολογίζουν το άθροισμα των στοιχείων ενός πίνακα Α με 10 στοιχεία;

A)	ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10 Σ ← Σ + Α ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	Γ)	ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10 Σ ← Σ + Α[Ι] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
B)	ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10 Σ ← Σ + Ι ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ	Δ)	ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10 Σ ← Α[Ι] ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΠΕΡΙΕΧΕΙ

- Θεωρία ανά κεφάλαιο
- Ερωτήσεις ανα κεφάλαιο Σωστού – Λάθους , Συμπλήρωσης Κενών, Αντιστοίχισης , Ανάπτυξης, ομαδοποιημένες ασκήσεις
- Λυμένες υποδειγματικές Ασκήσεις
- Άλυτες ασκήσεις