

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ



ΑΛΓΟΡΙΘΜΟΙ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ «ΓΛΩΣΣΑ»

Δομές Αλγορίθμων

Μέρος

1

Περιεχόμενα

1. Δομή ακολουθίας
2. Δομή επιλογής
3. Δομή επανάληψης

ΕΙΣΑΓΩΓΗ

ΕΠΕΞΗΓΗΣΗ

ΕΙΚΟΝΙΔΙΩΝ



Χρήσιμες πληροφορίες



Δοκιμάστε τις γνώσεις σας



Επανάληψη



Ιδιαίτερη Προσοχή

Στο πρώτο μέρος των σημειώσεων θα παρουσιαστεί η έννοια του αλγορίθμου, στο δεύτερο μέρος η έννοια των δομών δεδομένων και στο τρίτο μέρος θα παρουσιαστούν οι

βασικές αρχές του προγραμματισμού σε μια γλώσσα προγραμματισμού που είναι κατασκευασμένη για εκπαιδευτικούς σκοπούς, την «ΓΛΩΣΣΑ»

Για την σήμανση στις παραγράφους που πρέπει να παρατηρήσουμε κάτι το ξεχωριστό ή να γίνει σχετική επανάληψη, θα χρησιμοποιηθούν τα παραπάνω εικονίδια.

1. ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ

ΓΕΝΙΚΑ – ΠΑΡΑΔΕΙΓΜΑΤΑ

Το χαρακτηριστικό αυτής της δομής είναι ότι οι εντολές του αλγορίθμου εκτελούνται η μία μετά την άλλη. Έτσι η σειρά εκτέλεσης των εντολών είναι δεδομένη από την αρχή.

Οι εντολές εκτελούνται διαδοχικά, δηλαδή δεν μπορεί να εκτελεστεί η εντολή 5 αν δεν έχει προηγηθεί η εκτέλεση της εντολής 4.

Ο τόπος αυτός εκτέλεσης των εντολών πολλές φορές λέγεται «Σειριακή» ή «Ακολουθιακή» εκτέλεση των εντολών του αλγορίθμου.

Η δομή του αλγορίθμου ή τμήματος του που εκτελείται σειριακά ονομάζεται «Δομή ακολουθίας»

Παράδειγμα 1 Να γίνει αλγόριθμος που να διαβάζει δύο αριθμούς, να υπολογίζει και εκτυπώνει το άθροισμά τους

Βήματ	Αλγόριθμος	Παρατηρήσεις
1ο	Αλγόριθμος Άθροισμα	Αρχή αλγορίθμου
2ο	Διάβασε A	Εισαγωγή δεδομένων
3ο	Διάβασε B	
4ο	C ← A+B	Επεξεργασία δεδομένων
5ο	Εκτύπωσε C	Αποτελέσματα
6ο	Τέλος Άθροισμα	Τέλος αλγορίθμου

Οι λέξεις ή σύμβολα με χρώμα μπλε είναι οι εντολές του αλγορίθμου.

Κάθε αλγόριθμος ξεκινά με τη λέξη «Αλγόριθμος» και τελειώνει με τη λέξη «Τέλος»

Η λέξη «Άθροισμα» επιλέγεται από τον κατασκευαστή του αλγορίθμου που συνήθως είναι λέξη ή λέξεις που περιγράφουν την εργασία του αλγορίθμου. Είναι υποχρεωτική στην αρχή και στο τέλος του αλγορίθμου

Στο 2ο και 3ο Βήμα γίνεται **εισαγωγή** των δεδομένων με την εντολή «Διάβασε». Αποδίδει τιμές στις μεταβλητές A και B που δίδονται από τον χρήστη.

Στο 4ο Βήμα γίνεται η **επεξεργασία** των δεδομένων. Στην μεταβλητή C αποδίδεται το άθροισμα των τιμών των μεταβλητών A και B

Στο 5ο Βήμα έχουμε την έξοδο των **αποτελεσμάτων** με την εντολή «Εκτύπωσε» που εκτυπώνει την τιμή της μεταβλητής C

Για τον **έλεγχο της πληρότητας** του αλγορίθμου θα πρέπει να ικανοποιεί τα επόμενα κριτήρια :

Είσοδος (Input) οι μεταβλητές A και B

Έξοδος (Output) η μεταβλητή C

Καθοριστικότητα (Definiteness) Η κάθε εντολή εκτελεί εντελώς συγκεκριμένη ενέργεια.

Περατότητα (Finiteness) Μετά από έξι βήματα ο αλγόριθμος τελειώνει

Αποτελεσματικότητα (Effectiveness) κάθε εντολή είναι ιδιαίτερα σαφής και απλή

Παράδειγμα 2 Να δοθεί αλγόριθμος υπολογισμού του εμβαδού τραπεζίου, όταν δίνονται οι βάσεις και το ύψος του.

Βήματα	Αλγόριθμος	Παρατηρήσεις
1ο	Αλγόριθμος Εμβ_ Τραπεζίου	Αρχή αλγορίθμου
2ο	Διάβασε B1	Εισαγωγή δεδομένων
3ο	Διάβασε B2 Διάβασε Y	
4ο	$E \leftarrow (B1+B2)*Y/2$	Επεξεργασία δεδομένων
5ο	Εκτύπωσε E	Αποτελέσματα
6ο	Τέλος Εμβ_ Τραπεζίου	Τέλος αλγορίθμου

Παράδειγμα 3 Να γραφεί αλγόριθμος ο οποίος να διαβάσει την ακτίνα r και το ύψος h ενός κυλίνδρου, να υπολογίζει και να εμφανίζει τον όγκο (Volume) και το συνολικό εμβαδόν (Area) της επιφάνειας του σύμφωνα με τις σχέσεις.

$$\text{Volume} = \pi r^2 h \quad \text{Area} = 2\pi r h + 2\pi r^2$$

Αλγόριθμος Ογκος_Εμβ_Κυλίνδρου

Διάβασε r

Διάβασε h

$\text{Volume} \leftarrow 3,14*r^2*h$

$\text{Area} \leftarrow 2*3,14*r*h+2*3,14*r^2$

Εκτύπωσε Area

Εκτύπωσε Volume

Τέλος Ογκος_Εμβ_Κυλίνδρου

ΠΑΡΑΤΗΡΗΣΗ. Για την εντολή \leftarrow (απόδοσης τιμής) πρώτα γίνονται οι πράξεις στο 2ο μέλος και το αποτέλεσμα αποδίδεται στη μεταβλητή Area

Μια καλύτερη προσέγγιση στον παραπάνω αλγόριθμο είναι η παρακάτω:

Αλγόριθμος Ογκος_Εμβ_Κυλίνδρου2

! Εισαγωγή δεδομένων

Διάβασε r

Διάβασε h

! Υπολογισμός Ογκου και εμβαδού

$\text{Volume} \leftarrow 3,14*r^2*h$

$\text{Area} \leftarrow 2*3,14*r*h+2*3,14*r^2$

! Εκτύπωση αποτελεσμάτων

Εκτύπωσε Area

Εκτύπωσε Volume

Τέλος Ογκος_Εμβ_Κυλίνδρου2

Στη παραπάνω μορφή χρησιμοποιείται η εντολή «Σχόλιο» που συμβολίζεται με το «!» δεν επηρεάζει την εκτέλεση του αλγορίθμου, είναι βοηθητικό για τον κατασκευαστή του αλγορίθμου διότι έχει τη δυνατότητα να κρατά σημειώσεις μέσα στον αλγόριθμο. Για παράδειγμα στον παραπάνω αλγόριθμο ξεχωρίζουμε τις ενότητες : Εισαγωγή – Επεξεργασία – Αποτελέσματα.



ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΕΝΤΟΛΩΝ ΑΛΓΟΡΙΘΜΩΝ

1. **Αλγόριθμος** < Όνομα Αλγορίθμου >

2. **Τέλος** < Όνομα Αλγορίθμου >

3. **Διάβασε** < Όνομα Μεταβλητής >

Αποδίδει τιμή σε μεταβλητή κατά την εκτέλεση του αλγορίθμου ,κάνοντας ερώτημα στον χρήστη.

4. **Τύπωσε** < Όνομα μεταβλητής >

Τυπώνει την τιμή της μεταβλητής

5. **! Σχόλιο**

Σχόλιο –Δίπλα σε αυτό το σύμβολο μπορούμε να γράψουμε σχόλια , δεν επηρεάζει την εκτέλεση του αλγορίθμου

6. **← Εντολή εκχώρησης τιμής**

Απόδοση τιμής σε μεταβλητή κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Η σύνταξή της είναι :

Όνομα_Μεταβλητής ← έκφραση

Παραδείγματα :

A ← 132 : Η μεταβλητή A παίρνει την τιμή 132

Embado ← Bash*ypso : Εκτελούνται οι πράξεις στο δεύτερο μέλος και το αποτέλεσμα αποδίδεται στη μεταβλητή Embado

ΠΡΑΞΕΙΣ-ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΠΡΑΞΕΩΝ

Στα παραπάνω παραδείγματα παρατηρούμε ότι κατά τη διαδικασία της επεξεργασίας των δεδομένων εκτελούνται διάφορες πράξεις μεταξύ των τιμών των μεταβλητών. Οι πράξεις που χρησιμοποιούμε μεταξύ των μεταβλητών στους αλγορίθμους είναι αυτές που μπορεί να εκτελέσει ο Η/Υ και είναι οι παρακάτω

Πράξη	Σύμβ	Παρατηρ
Πρόσθεση	+	
Αφαίρεση	-	
Πολλαπλασιασμός	*	
Διαίρεση	/	
Ύψωση σε δύναμη	^	
Ακέραιο Πηλίκο δύο ακεραίων	\ ή	
Ακέραιο υπόλοιπο	.. Mod	

Η προτεραιότητα των πράξεων είναι η ίδια με την άλγεβρα δηλαδή :

Πρώτα εκτελούνται οι πράξεις μέσα στις παρενθέσεις

Εκτέλεση των δυνάμεων

Οι Πολλαπλασιασμοί και οι διαιρέσεις

Προσθέσεις και αφαιρέσεις



ΑΣΚΗΣΕΙΣ Στη Δομή της Ακολουθίας

Άσκηση 1 Να γραφεί αλγόριθμος που να υπολογίζει τις τιμές της πιο κάτω παράστασης :

$$E1 = \frac{(B + C)^4}{C - A} \quad \text{για } A=2, B=1, C=4$$

Άσκηση 2 Να γραφεί αλγόριθμος ο οποίος να διαβάζει τις ακεραίες μεταβλητές A, B, C, D να υπολογίζει και να εμφανίζει το άθροισμα τους SUM, το γινόμενο τους PRODUCT και το μέσο όρο τους AVER

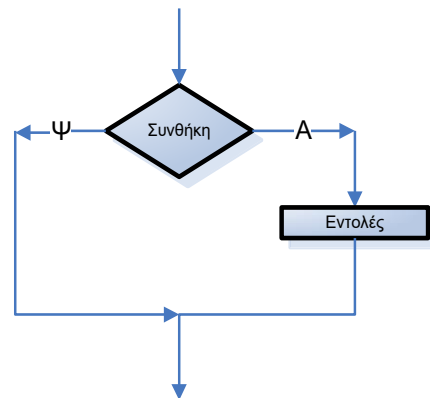
2. ΔΟΜΗ ΕΠΙΛΟΓΗΣ

Με την διαδικασία της σειριακής /ακολουθιακής εκτέλεσης των εντολών σ' ένα αλγόριθμο πολύ λίγα προβλήματα μπορεί να λυθούν. Στα περισσότερα προβλήματα σε κάποιο σημείο της διαδικασίας εκτέλεσης του αλγορίθμου θα χρειαστεί να πάρουμε κάποια απόφαση. Για παράδειγμα να συγκρίνουμε δύο αριθμούς, και ανάλογα με τη σχέση τους να ακολουθήσουμε τον κατάλληλο δρόμο στη πορεία εκτέλεσης του αλγορίθμου. Το χαρακτηριστικό της δομής Επιλογής είναι ότι υπάρχει κάποια συνθήκη. Η συνθήκη αυτή μπορεί να χαρακτηριστεί «Αληθής» ή «Ψευδής» ανάλογα με την τιμή αυτής της συνθήκης συνεχίζει η ροή του αλγορίθμου. Υπάρχουν τριών ειδών δομές επιλογής. Παρακάτω θα μελετήσουμε την κάθε μια χωριστά με παραδείγματα και επεξηγήσεις.

1η Μορφή

Αν <συνθήκη> **τότε**
ομάδα εντολών
τέλος_αν

Διάγραμμα Ροής =====>



Η μορφή αυτή χρησιμοποιεί τις λέξεις **Αν** –**τότε** –**Τέλος_αν**. Αν η συνθήκη είναι αληθής τότε εκτελείται η ομάδα εντολών που βρίσκεται ανάμεσα στις εντολές **Αν ... τότε** – **Τέλος_αν**

Παράδειγμα 1. Να γίνει αλγόριθμος που να διαβάζει ένα αριθμό και να εκτυπώνει την απόλυτη τιμή του

Αλγόριθμος Απόλυτη_τιμή

Διάβασε Ar

Αν Ar < 0 **τότε**

Ar ← Ar*(-1)

Τέλος_αν

Τύπωσε Ar

Τέλος Απόλυτη_τιμη

Κατ' αρχήν ο αλγόριθμος δέχεται σαν εισοδο ένα αριθμό A_r .

Αν ο αριθμός είναι αρνητικός τότε εκτελείται η εντολή $A_n \dots$ τότε $- \text{τελος_αν}$ και αποδίδεται στη μεταβλητή A_r η τιμή $-A_r$. Μετά γίνεται η εκτύπωση του αριθμού A_r που είναι πλέον θετικός.

Αν ο αριθμός A_r που δίνουμε είναι θετικός τότε παρακάμπτεται η εντολή $A_n \dots$ τότε $- \text{Τέλος_αν}$ και απλώς γίνεται η εκτύπωση του αριθμού

Παράδειγμα 2 Να γραφεί αλγόριθμος ο οποίος να διαβάζει το βαθμό μαθήματος ενός μαθητή. Αν ο βαθμός είναι μεγαλύτερος ή ίσος του 10 να εμφανίζει το μήνυμα ότι πέρασε το μάθημα

Αλγόριθμος Βαθμός

Διάβασε B

Αν $B \geq 10$ **τότε**

Τύπωσε «Πέρασε το μάθημα»

Τέλος_αν

Τέλος Βαθμός

Αν το προηγούμενο πρόβλημα μας ζητούσε να τυπώνει ότι «κόπηκε στο μάθημα» στη περίπτωση που $B < 10$ τότε θα έπρεπε ο προηγούμενος αλγόριθμος να γραφεί όπως παρακάτω :

Αλγόριθμος Βαθμός

Διάβασε B

Αν $B \geq 10$ **τότε**

Τύπωσε «Πέρασε το μάθημα»

Τέλος_αν

Αν $B < 10$ **τότε**

Τύπωσε «Κόπηκε στο μάθημα»

Τέλος_αν

Τέλος Βαθμός

Τέτοια προβλήματα μπορούν να αντιμετωπιστούν καλύτερα με τη 2^η μορφή επιλογής

2η Μορφή

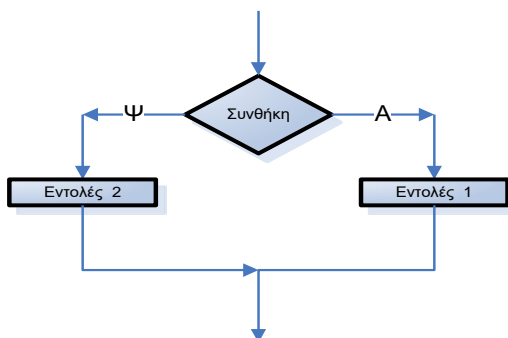
Αν <συνθήκη> **τότε**

Ομάδα εντολών 1

αλλιώς

Ομάδα εντολών 2

τέλος_αν



Διάγραμμα ροής =====>

Στη παραπάνω περίπτωση η εντολή εκτελείται ως εξής :Αρχικά ελέγχεται η συνθήκη. Στη συνέχεια υπάρχουν δύο δυνατές περιπτώσεις

Αν η συνθήκη είναι «αληθής» (Ικανοποιείται), εκτελείται μόνο η ομάδα εντολών 1. Η ομάδα εντολών 2 αγνοείται και ο έλεγχος του αλγόριθμου βγαίνει από τη δομή επιλογής.

Σε αντίθετη περίπτωση, αν δηλαδή η συνθήκη είναι «Ψευδής» (δεν ικανοποιείται), δε λαμβάνεται υπόψη η ομάδα εντολών 1. Εκτελείται μόνο η ομάδα εντολών 2 και η εκτέλεση της δομής επιλογής τελειώνει.

Παράδειγμα 1 Να γραφεί αλγόριθμος ο οποίος να διαβάσει δύο αριθμούς A και B να τους συγκρίνει και να εμφανίζει ποιος είναι μεγαλύτερος

Αλγόριθμος Σύγκριση

Διάβασε A

Διάβασε B

Αν $A > B$ **τότε**

Τύπωσε «Ο αριθμός A είναι μεγαλύτερος του B»

Αλλιώς

Τύπωσε «Ο αριθμός B είναι μεγαλύτερος του A»

Τέλος_αν

Τέλος Σύγκριση

Βέβαια ο παραπάνω αλγόριθμος δεν εξετάζει την περίπτωση που ο αριθμός A είναι ίσος με τον B

Παράδειγμα 2 Να γραφεί αλγόριθμος ο οποίος να διαβάσει ένα αριθμό και να δίνει την απόλυτη τιμή του αριθμού αυτού

Αλγόριθμος Απόλυτη_Τιμή

Διάβασε A

Αν $A > 0$ **τότε**

B \leftarrow A

Αλλιώς

B \leftarrow -A

Τέλος_αν

Τύπωσε B

Τέλος Απόλυτη_Τιμή

Εδώ παρατηρούμε ότι στη περίπτωση που ο αριθμός A είναι θετικός αποδίδεται στη μεταβλητή B ο αριθμός A, διαφορετικά αποδίδεται στη μεταβλητή B ο αντίθετος του A (Θετικός). Έτσι σε κάθε περίπτωση η μεταβλητή B έχει τιμή ίση με την απόλυτη τιμή της μεταβλητής A

Παράδειγμα 3 Να γραφεί αλγόριθμος ο οποίος να διαβάσει ένα ακέραιο αριθμό N. Να ελέγχει αν ο αριθμός είναι άρτιος και να εμφανίζει ένα μήνυμα ότι ο αριθμός είναι άρτιος, διαφορετικά να εμφανίζει μήνυμα ότι ο αριθμός είναι περιττός

Μαθηματική λύση: Στα μαθηματικά για να ελέγξουμε αν ένας αριθμός είναι άρτιος ή περιττός τον διαιρούμε με το 2.

Αν το υπόλοιπο της διαίρεσης είναι μηδέν τότε ο αριθμός είναι άρτιος ($n=2k$), ενώ αν το υπόλοιπο είναι 1 τότε ο αριθμός είναι περιττός ($n=2k+1$)

Στους αλγορίθμους χρησιμοποιούμε τον τελεστή **Mod** για να ελέγξουμε το υπόλοιπο μιας ακέραιας διαίρεσης με το 2. Για παράδειγμα :

H έκφραση $7 \bmod 2$ δίνει την τιμή 1, που είναι το υπόλοιπο της διαίρεσης του 7 με το 2.

H έκφραση $8 \bmod 2$ δίνει την τιμή 0, που είναι το υπόλοιπο της διαίρεσης του 8 με το 2

Αλγόριθμος Άρτιος_Περιττός

Διάβασε A

$Y_{\text{rol}} \leftarrow A \bmod 2$
Αν $Y_{\text{rol}}=0$ **τότε**
 Τύπωσε “ Ο Αριθμός είναι άρτιος»
Αλλιώς
 Τύπωσε “ Ο Αριθμός είναι Περιττός»
Τέλος_αν
Τέλος Άρτιος_Περιττός

Στον παραπάνω αλγόριθμο η τιμή της μεταβλητής y_{rol} είναι το υπόλοιπο της διαίρεσης του αριθμού A με το 2.

Το αποτέλεσμα του αλγορίθμου είναι το μήνυμα «Ο Αριθμός A είναι άρτιος-Περιττός».

Στο παραπάνω πρόβλημα θα μπορούσε να χρησιμοποιηθεί μια μεταβλητή που να δεχτεί τιμή το αποτέλεσμα του αλγορίθμου (π.χ $\text{apotel} \leftarrow$ «Ο Αριθμός A είναι άρτιος»). Δείτε μια άλλη λύση του προηγούμενου προβλήματος :

Αλγόριθμος Άρτιος_Περιττός2

! Εισαγωγή δεδομένων

Διάβασε A

! Επεξεργασία δεδομένων

$Y_{\text{rol}} \leftarrow A \bmod 2$

Αν $Y_{\text{rol}}=0$ **τότε**

$\text{apotel} \leftarrow$ “ Ο Αριθμός A είναι άρτιος»

Αλλιώς

$\text{apotel} \leftarrow$ “ Ο Αριθμός A είναι Περιττός»

Τέλος_αν

! Αποτελέσματα

Τύπωσε apotel

Τέλος Άρτιος_Περιττός2

Περισσότερα για την εντολή Απόδοσης τιμής «←»

Μέχρι τώρα είχαμε συνηθίσει εκφράσεις απόδοσης αριθμητικής τιμής σε μεταβλητή όπως για παράδειγμα: $\text{met} \leftarrow 4$, $B \leftarrow A$, $Y_{\text{rol}} \leftarrow A \bmod 2$ στο τελευταίο παράδειγμα ($\text{apotel} \leftarrow$ “ Ο Αριθμός A είναι Περιττός») βλέπουμε απόδοση μιας πρότασης – ενός συνόλου χαρακτήρων στην μεταβλητή apotel . Δηλαδή έχουμε δικαίωμα να αποδώσουμε ένα string (σύνολο χαρακτήρων) σε μια μεταβλητή.

3^η Μορφή

Επίλεξε έκφραση

Περίπτωση 1

 Ομάδα εντολών 1

Περίπτωση 2

 Ομάδα εντολών 2

Περίπτωση ν

 Ομάδα εντολών ν

Περίπτωση αλλιώς

Ομάδα εντολών

Τέλος_επιλογών

Αν οι διαφορετικές επιλογές είναι περισσότερες των δύο είναι προτιμότερο να χρησιμοποιείται το παραπάνω σχήμα (3^η μορφή) που είναι ένα σχήμα πολλαπλής επιλογής.
☼ Η εντολή χρησιμοποιεί τις βοηθητικές λέξεις (**επίλεξε, περίπτωση, περίπτωση αλλιώς και τέλος_επιλογών**).

☼ Αρχίζει με τη βοηθητική λέξη **επίλεξε** και τελειώνει με τη βοηθητική λέξη **τέλος_επιλογών**.
☼ Μετά το **επίλεξε** ακολουθεί η μεταβλητή.
Ακολουθεί η λέξη **περίπτωση**, ακολουθούμενη από **μία από τις πιθανές τιμές της μεταβλητής ή διακριτές τιμές ή κάποια συνθήκη ή μια περιοχή τιμών**

Παράδειγμα 1 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα αιέριο αριθμό N από το 1 μέχρι το 7 και να εμφανίζει την αντίστοιχη ημέρα της εβδομάδας (1 για Κυριακή, 2 για Δευτέρα, ..., 7 για Σάββατο)

Αλγόριθμος Μερα_Εβδομάδας

! Εισαγωγή δεδομένων

Γράψε «Δώσε ένα αριθμό από 1 έως 7»

Διάβασε Number

! Επεξεργασία δεδομένων

Επίλεξε Number

Περίπτωση 1

Mera ← «ΚΥΡΙΑΚΗ»

Περίπτωση 2

Mera ← «ΔΕΥΤΕΡΑ»

Περίπτωση 3

Mera ← «ΤΡΙΤΗ»

Περίπτωση 4

Mera ← «ΤΕΤΑΡΤΗ»

Περίπτωση 5

Mera ← «ΠΕΜΠΤΗ»

Περίπτωση 6

Mera ← «ΠΑΡΑΣΚΕΥΗ»

Περίπτωση 7

Mera ← «ΣΑΒΒΑΤΟ»

Περίπτωση αλλιώς

Mera ← «ΕΔΩΣΕΣ ΛΑΘΟΣ ΑΡΙΘΜΟ»

Τέλος_επιλογών

! Αποτελέσματα

Τύπωσε Mera

Τέλος Μερα_Εβδομάδας

Αφού τυπωθεί το μήνυμα με την εντολή **Γράψε «Δώσε ένα αριθμό από 1 έως 7»** η επόμενη διαβάζει την τιμή και την θέτει στη μεταβλητή Number, ανάλογα με την τιμή που δώσαμε η μεταβλητή Mera παίρνει την αντίστοιχη τιμή. Μετά το τέλος επιλογών τυπώνεται η τιμή της μεταβλητής Mera.

- ΠΑΡΑΤΗΡΗΣΕΙΣ**
- 1) Τη λέξη «Περίπτωση» την ακολουθούν **πιθανές** τιμές.
 - 2) Η μεταβλητή Mera δέχεται τιμή String
 - 3) Το τμήμα Εισαγωγή δεδομένων στον αλγόριθμο είναι βελτιωμένο σε σχέση με προηγούμενα. Αφού τυπωθεί το σχετικό μήνυμα ακολουθεί η εισαγωγή της τιμής της μεταβλητής Number

Παράδειγμα 2 (Με διακριτή τιμή) Να γραφεί αλγόριθμος ο οποίος να διαβάζει δύο ακέραιους αριθμούς X, Y. Θα διαβάζει επίσης έναν από τους τρεις χαρακτήρες (+, -, *). Ανάλογα με το χαρακτήρα που διαβάζει θα κάνει και την αντίστοιχη πράξη (πρόσθεση, αφαίρεση ή πολλαπλασιασμό) και θα εμφανίζει τα κατάλληλα μηνύματα

Αλγόριθμος Πράξεις

Διάβασε A

Διάβασε B

Διάβασε SHMA

Επίλεξε SHMA

Περίπτωση «+»

APOT ← A+B

Περίπτωση «-»

APOT ← A-B

Περίπτωση «*»

APOT ← A*B

Περίπτωση Αλλιώς

Τύπωσε «Λάθος πράξη»

Τέλος_Επιλογών

Τύπωσε APOT

Τέλος Πράξεις

ΠΑΡΑΤΗΡΗΣΗ. Η διακριτή τιμή είναι ένα αλφαριθμητικό «+» ή «-» ή «*» που αποδίδεται στη μεταβλητή APOT

Παράδειγμα 3 (Με συνθήκη) Να γίνει αλγόριθμος που να χαρακτηρίζει την επίδοση ενός μαθητή και συγκεκριμένα : Αν η βαθμολογία του είναι μικρότερη του 10 – χαρακτηρισμός «ΑΠΟΡΡΙΠΤΕΤΑΙ», Αν η βαθμολογία είναι μεγαλύτερη ίση του 10 – χαρακτηρισμός «ΠΡΟΑΓΕΤΑΙ» (με τη 3η μορφή πολλαπλών επιλογών)

Αλγόριθμος Χαρακτηρισμός

Διάβασε B

Επίλεξε B

Περίπτωση <0

Τύπωσε «Είπαμε βαθμός»

Περίπτωση <10

Τύπωσε «ΑΠΟΡΡΙΠΤΕΤΑΙ»

Περίπτωση >10

Τύπωσε «ΠΡΟΑΓΕΤΑΙ»

Περίπτωση >20

Τύπωσε «ΠΑΡΑ ΕΙΝΑΙ ΑΡΙΣΤΟΥΧΟΣ»

Τέλος_Επιλογών

Τέλος Χαρακτηρισμός

ΠΑΡΑΤΗΡΗΣΗ. Η μεταβλητή B ικανοποιεί την συνθήκη (<0) ή (<10) ή (>10) ή (>20)

Παράδειγμα 4 (Με περιοχή τιμών) Να γίνει αλγόριθμος που να χαρακτηρίζει την επίδοση ενός μαθητή και συγκεκριμένα : Αν η βαθμολογία του είναι μικρότερη του 10 – χαρακτηρισμός «ΑΠΟΡΡΙΠΤΕΤΑΙ», Αν η βαθμολογία είναι από 10 μέχρι 13 – χαρακτηρισμός «ΜΕΤΡΙΑ» από 14 μέχρι 16 –Χαρακτηρισμός «ΚΑΛΑ» από 17 μέχρι 20 – Χαρακτηρισμός «ΑΡΙΣΤΑ» (με τη 3η μορφή πολλαπλών επιλογών)

Αλγόριθμος Χαρακτηρισμός2

Διάβασε B

Επίλεξε B

Περίπτωση από 1 μέχρι 9

Τύπωσε «ΑΠΟΡΡΙΠΤΕΤΑΙ»

Περίπτωση από 10 μέχρι 13

Τύπωσε «ΜΕΤΡΙΑ»

Περίπτωση Από 14 μέχρι 16

Τύπωσε «ΚΑΛΑ»

Περίπτωση Από 17 μέχρι 20

Τύπωσε «ΑΡΙΣΤΑ»

Περίπτωση Αλλιώς

Τύπωσε «ΕΠΙΠΑΜΕ ΒΑΘΜΟΣ !!!!»

Τέλος_Επιλογών

Τέλος Χαρακτηρισμός2

ΠΑΡΑΤΗΡΗΣΗ. Η μεταβλητή B βρίσκεται στη περιοχή τιμών (1-9) ή 10-13) ή (14-16) ή (17-20) ή (>20) ή (<1)

4^η Μορφή – παραλλαγή της 3^{ης} μορφής επιλογής

Αν <συνθήκη 1> τότε

Ομάδα εντολών 1

αλλιώς_αν <συνθήκη 2> τότε

Ομάδα εντολών 2

αλλιώς_αν <συνθήκη 3 > τότε

Ομάδα εντολών 3

.....

αλλιως

Ομάδα εντολών ν

τέλος_αν

Αν λάβουμε υπ' όψιν μας την 4^η μορφή το παράδειγμα 1 μπορεί να γραφεί :

Αλγόριθμος Πράξεις2

Διάβασε A

Διάβασε B

Διάβασε SHMA

άν SHMA = '+' τότε

ΑΡΟΤ ← A+B

Αλλιώς_αν SHMA = '-' τότε

ΑΡΟΤ ← A-B

Αλλιώς_αν SHMA = '*' τότε

ΑΡΟΤ ← A*B

Αλλιώς

Τύπωσε «Λάθος πράξη»



ΑΣΚΗΣΕΙΣ ΣΤΗΝ ΔΟΜΗ ΤΗΣ Ακολουθίας- Επιλογής

ΑΣΚΗΣΗ 1 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα ακέραιο αριθμό N. Να ελέγχει αν ο αριθμός είναι μεγαλύτερος ή ίσος του 0 και να εμφανίζει ένα μήνυμα ότι ο αριθμός είναι θετικός

ΑΣΚΗΣΗ 2 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα ακέραιο αριθμό N. Να ελέγχει αν ο αριθμός είναι μεγαλύτερος ή ίσος του 0 και να εμφανίζει ένα μήνυμα ότι ο αριθμός είναι θετικός, διαφορετικά αν ο αριθμός είναι μικρότερος του 0 να εμφανίζει μήνυμα ότι ο αριθμός είναι αρνητικός.

ΑΣΚΗΣΗ 3 Να γραφεί αλγόριθμος ο οποίος να διαβάζει τα αποτελέσματα των εξετάσεων αγγλικών ενός μαθητή. Αν ο μαθητής πήρε A να εμφανίζει το μήνυμα ΑΡΙΣΤΑ, B το μήνυμα ΚΑΛΑ, C το μήνυμα ΜΕΤΡΙΑ και D το μήνυμα ΑΠΕΤΥΧΕ

Άσκηση 4 Τι θα εκτυπώσουν τα παρακάτω τμήματα αλγορίθμων ;

$A \leftarrow 5$ Met \leftarrow 'A' Γράψε Met	$A \leftarrow 5$ Met \leftarrow A Γράψε Met
---	---

Άσκηση 5 Τι θα τυπώσουν οι παρακάτω εντολές ;
 Γράψε $12*3$, Γράψε '12*3'

Άσκηση 6 Τι θα εμφανίσει καθένα από τα ακόλουθα τμήματα αλγορίθμων;

A	B	Γ	Δ
$X \leftarrow 5$ αν $(x > 5)$ τότε $x \leftarrow x + 4$ αλλιώς $x \leftarrow x - 4$ τέλος_αν εμφάνισε x	$X \leftarrow 7$ αν $(x > 5)$ τότε $x \leftarrow x + 4$ αλλιώς $x \leftarrow x - 4$ τέλος_αν εμφάνισε x	$X \leftarrow 5$ αν $(x \geq 5)$ τότε $x \leftarrow x + 4$ τέλος_αν αν $(x < 5)$ τότε $x \leftarrow x - 4$ τέλος_αν εμφάνισε x	$x \leftarrow 7$ αν $(x > 5)$ τότε $x \leftarrow x - 4$ τέλος_αν αν $(x < 5)$ τότε $x \leftarrow x + 6$ τέλος_αν εμφάνισε x

Άσκηση 7 Να γίνει πρόγραμμα που να ζητάει 2 ακέραιες τιμές τις οποίες θα αποθηκεύει σε 2 διαφορετικές μεταβλητές. Στη συνέχεια να εμφανίζει τη μεγαλύτερη από αυτές

Άσκηση 8 γίνε πρόγραμμα που να ζητάει 3 ανέραίες τιμές τις οποίες θα αποθηκεύει σε 3 διαφορετικές μεταβλητές. Στη συνέχεια να εμφανίζει τη μεγαλύτερη από αυτές

Άσκηση 9 Να γράψεις έναν αλγόριθμο που διαβάσει τις δύο πλευρές ενός ορθογωνίου και θα υπολογίζει και θα εμφανίζει το εμβαδόν του μόνο αν και οι δύο πλευρές του είναι θετικοί αριθμοί, σε αντίθετη περίπτωση θα τυπώνει κατάλληλο μήνυμα. ($E = a * b$).

Άσκηση 10 Να γράψεις έναν αλγόριθμο που θα διαβάσει από το πληκτρολόγιο έναν φυσικό αριθμό και ανάλογα αν ο αριθμός είναι από 1 έως 12 να εμφανίζει με γράμματα το μήνα που αντιστοιχεί στον αριθμό αυτόν (πχ. 1 → ΙΑΝΟΥΑΡΙΟΣ 2 → ΦΕΒΡΟΥΑΡΙΟΣ ... 12 → ΔΕΚΕΜΒΡΙΟΣ), ενώ αν ο αριθμός είναι μεγαλύτερος από 12 και μικρότερος από 1 να τυπώνεται στην οθόνη του υπολογιστή το μήνυμα "ΛΑΘΟΣ ΔΕΔΟΜΕΝΑ".

Άσκηση 11 Να γράψεις έναν αλγόριθμο που θα διαβάσει από το πληκτρολόγιο ένα λεκτικό που θα εισάγει ο χρήστης. Μετά την εισαγωγή το λεκτικό θα συγκρίνεται με ένα προκαθορισμένο λεκτικό κλειδί (πχ '12345') και αν τα δύο λεκτικά είναι τα ίδια τότε θα τυπώνεται στην οθόνη του υπολογιστή το μήνυμα "ΚΩΔΙΚΟΣ ΣΩΣΤΟΣ" ενώ αν τα δύο λεκτικά είναι διαφορετικά τότε να τυπώνεται στην οθόνη του υπολογιστή το μήνυμα "ΛΑΘΟΣ ΚΩΔΙΚΟΣ".

Άσκηση 12 Να γίνει τμήμα αλγορίθμου που εκτελεί αντιμετάθεση των τιμών δύο μεταβλητών

Άσκηση 13 Να γίνει αλγόριθμος που θα διαβάσει ένα χρηματικό ποσό σε δραχμές και θα υπολογίζει και θα εμφανίζει το αντίστοιχο ποσό σε €.

Άσκηση 14 Σε μια αλυσίδα καταστημάτων οι τιμές των προϊόντων στις ετικέτες αναγράφονται χωρίς το ΦΠΑ. Να γίνει ο αλγόριθμος που εκτελεί η ταμιακή μηχανή, ώστε να υπολογίζει το πληρωτέο ποσό με το ΦΠΑ, καθώς και τα ρέστα που δικαιούται ο πελάτης (ΦΠΑ 18%).

Άσκηση 15 Η εταιρία ΑΑΑ έχει 3 υποκαταστήματα. Να γίνει αλγόριθμος που θα διαβάσει τα έσοδα για κάθε υποκατάστημα και θα εμφανίζει τα συνολικά έσοδα της εταιρίας, καθώς και το ποσοστό συμμετοχής σε αυτά για καθένα από τα 3 υποκαταστήματα.

Άσκηση 16 Το επίδομα που δίνει μία εταιρία στους εργατες το καλοκαίρι φαίνεται παρακάτω :

Αριθμός Παιδιών	Επίδομα ως ποσοστό επί του μισθού
1	10%
2	18%
3+	27%

Να γίνει αλγόριθμος που θα διαβάσει το μισθό και τον αριθμό των παιδιών ενός εργατή και θα εμφανίζει το σύνολο του μισθού με το επίδομα.

ΛΟΓΙΚΕΣ ΠΡΟΤΑΣΕΙΣ- ΣΥΝΘΕΤΕΣ ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

Μια πρόταση που μπορεί να χαρακτηριστεί «Αληθής» ή «Ψευδής» θα ονομάζεται **λογική πρόταση**. Για παράδειγμα προτάσεις όπως

Προτάσεις	Χαρακτηρισμ
$120 > 7$	Α
$30 < 120$	Ψ
«Σήμερα χιονίζει»	Α ή Ψ
$A = 20$	Α ή Ψ
$X >= 80$	Α ή Ψ

Είναι λογικές προτάσεις γιατί μπορούν να χαρακτηριστούν «Αληθείς» ή «Ψευδείς» Όπως στις αριθμητικές πράξεις χρησιμοποιούμε τους αριθμητικούς τελεστές «+», «-», «*», «/» και κατασκευάζουμε συνθετότερες αριθμητικές ή αλγεβρικές παραστάσεις, έτσι και με τις λογικές πράξεις μπορούμε να χρησιμοποιήσουμε τους λογικούς τελεστές Η, ΚΑΙ, ΟΧΙ

Λογική Πράξη	Σύμβολο	Εναλλακτικό σύμβολο
Διάζευξη	H	OR
Συζευξη	KAI	AND
Αρνηση	OXI	NOT

Για να κατασκευάσουμε συνθετότερες λογικές προτάσεις, που πολλές φορές είναι χρήσιμες για τη δημιουργία συνθηκών οι οποίες χρησιμοποιούνται στις εντολές επιλογής.

Λογική πράξη «Αρνηση»

Q	OXI Q
Α	Ψ
Ψ	Α

Παράδειγματα : **OXI** ($7 > 8$) είναι Αληθής
OXI ($A = B$) αν $A = 5$ και $B = 12$ είναι Αληθής
OXI ($A + B > 10$) Αν $A = 5$ και $B = 7$ είναι Ψευδής

Λογική πράξη «Συζευξη»

Q	P	Q ΚΑΙ P
Α	Α	Α
Α	Ψ	Ψ
Ψ	Α	Ψ
Ψ	Ψ	Ψ

Όπως φαίνεται από τον παραπάνω πίνακα δύο λογικές προτάσεις Q , P που συνδέονται με το λογικό ΚΑΙ δημιουργούν με νέα λογική πρόταση (Q ΚΑΙ P) η οποία είναι **αληθής** μόνο στην περίπτωση που **και οι δύο** προτάσεις που τη συνθέτουν είναι **Αληθείς**. Σε όλες τις άλλες περιπτώσεις η σύνθετη πρόταση είναι ψευδής

Παραδείγματα : (7<8) **ΚΑΙ** (6<12) Αληθής
(A=15) **ΚΑΙ** (B<20) Αν A=6 , B=10 είναι **Ψευδής**

Λογική πράξη «Διάζευξη»

Q	P	Q Η P
A	A	A
A	Ψ	A
Ψ	A	A
Ψ	Ψ	Ψ

Όπως φαίνεται από τον παραπάνω πίνακα δύο λογικές προτάσεις Q , P που συνδέονται με το λογικό Η δημιουργούν με νέα λογική πρόταση (Q Η P) η οποία είναι **Ψευδής** μόνο στην περίπτωση που **και οι δύο** προτάσεις που τη συνθέτουν είναι **Ψευδείς**. Σε όλες τις άλλες περιπτώσεις η σύνθετη πρόταση είναι **Αληθής**

Παραδείγματα : (7>8) **Η** (6<12) Αληθής
(A=15) **Η** (B<20) Αν A=6 , B=10 είναι Αληθής



ΑΣΚΗΣΕΙΣ επαναληπτικές

Άσκηση 1 Να γίνει αλγόριθμος που να υπολογίζει την αριθμητική τιμή της παράστασης

$$A = \frac{X^2 + 5X}{\sqrt{X}}$$

Άσκηση 2 Αν το A έχει την τιμή 10 και το B την τιμή 20 τότε η έκφραση (A>8 ΚΑΙ B<20) Η (A>10 Η B=10) είναι αληθής ή ψευδής

Άσκηση 3 Να γράψεις έναν αλγόριθμο που θα διαβάζει από το πληκτρολόγιο έναν πραγματικό αριθμό ar. Ο αλγόριθμος θα εξετάζει αν ο αριθμός αυτός είναι διψήφιος ή όχι και ανάλογα να εμφανίζει κατάλληλο μήνυμα

Άσκηση 4 Να γίνει αλγόριθμος που θα διαβάζει το βαθμό ενός μαθητή και θα εμφανίζει τον αντίστοιχο χαρακτηρισμό, βάσει των παρακάτω :

Βαθμός	Χαρακτηρισμός
B>=18	Άριστα
15<=B<18	Πολύ καλά
12<=B<15	Καλά
9<=B <12	Μέτρια

$B < 9$	Ανεπαρκώς
---------	-----------

Άσκηση 5 Να γίνει αλγόριθμος που θα διαβάζει το βάρος ενός ατόμου και θα εμφανίζει ανάλογο μήνυμα, με βάση τον παρακάτω πίνακα :

ΒΑΡΟΣ	ΜΗΝΥΜΑ
$B \geq 100$	Χάστε επειγόντως βάρος!!
$100 > B \geq 90$	Πρέπει να χάσετε βάρος...
$90 > B \geq 80$	Έχετε κανονικό βάρος
$80 > B \geq 70$	Είστε αδύνατος
$B < 70$	Πρέπει να πάρετε κιλά!!!

ΕΜΦΩΛΕΥΜΕΝΗ ΕΠΙΛΟΓΗ

Οι εντολές που περιέχονται σε μία ομάδα εντολών μιας δομής επιλογής δεν είναι κατ' ανάγκη απλές (ακολουθιακές). Είναι δυνατό η επίλυση ενός προβλήματος να απαιτεί σε μια τέτοια ομάδα εντολών να περιέχονται άλλες εντολές επιλογής. Πολύ συχνά μια δομή επιλογής μπαίνει μέσα σε μια άλλη δομή επιλογής κ.λ.π. Δείτε την παρακάτω μορφή

```

Αν <συνθήκη1> τότε
    Ομάδα εντολών 1
Αν <συνθήκη2> τότε
    Ομάδα εντολών 2
τέλος_αν
αλλιώς
    Ομάδα εντολών 2
Αν <συνθήκη> τότε
    Ομάδα εντολών 3
αλλιώς
    ομάδα εντολών 4
τέλος_αν
τέλος_αν

```

ΑΣΚΗΣΕΙΣ ΣΤΗΝ ΕΜΦΩΛΕΥΜΕΝΗ ΕΠΙΛΟΓΗ

Άσκηση 1 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει το μηνιαίο λογαριασμό του νερού ως εξής: θα διαβάζει από το πληκτρολόγιο το κόστος για την χρήση αποχέτευσης σε

δραχμές, το κόστος για την συντήρηση του δικτύου σε δραχμές και την κατανάλωση του νερού σε κυβικά. Τα πρώτα 20 κυβικά χρεώνονται με 60 δραχμές το κυβικό και πάνω από τα 20 κυβικά χρεώνονται με 150 δραχμές το κυβικό

Άσκηση 2 Να γραφεί αλγόριθμος στον οποίο θα δίδονται τρεις αριθμοί και θα υπολογίζει τον μέγιστο απ' αυτούς

Άσκηση 3 Να γραφεί αλγόριθμος στον οποίο θα δίδονται τρεις αριθμοί και θα υπολογίζει τον ελάχιστο απ' αυτούς

Άσκηση 4 Να γραφεί αλγόριθμος ο οποίος να διαβάσει τρεις αριθμούς από το πληκτρολόγιο, να υπολογίζει και να τυπώνει το αποτέλεσμα του γινομένου του μεγαλύτερου από τους τρεις αριθμούς με το αποτέλεσμα της αφαίρεσης των άλλων δύο, υπό την προϋπόθεση ότι από τους δύο αυτούς θα αφαιρούμε πάντα το μικρότερο από το μεγαλύτερο

ΑΣΚΗΣΗ 5 Να γραφεί αλγόριθμος ο οποίος να αποφασίζει αν ένας μαθητής προβιβάστηκε ή όχι σύμφωνα με τα αποτελέσματα των γραπτών του σε 4 μαθήματα. Ο μαθητής προβιβάζεται αν ο μέσος όρος των βαθμών του στα 4 μαθήματα είναι τουλάχιστον 10 υπό την προϋπόθεση ότι δεν πήρε σε κανένα μάθημα βαθμό κάτω από 8.

Άσκηση 6 Να γραφεί αλγόριθμος που υπολογίζει και εμφανίζει τη λύση (ή τις λύσεις) μιας εξίσωσης δευτέρου βαθμού ($ax^2+bx+c = 0$) όταν είναι γνωστοί οι συντελεστές του δευτεροβάθμιου (a), πρωτοβάθμιου (b) και σταθερού της όρου (c).

Άσκηση 7 Να γραφεί αλγόριθμος που υπολογίζει και εμφανίζει την τιμή της παρακάτω συνάρτησης πολλαπλού τύπου για κάποιο x που ανήκει στο σύνολο των πραγματικών αριθμών:

$$f(x) = \begin{cases} x-1 & \text{για } x < 0 \\ x^2-3x & \text{για } 0 \leq x \leq 1 \\ x+1 & \text{για } x > 1 \end{cases}$$

Άσκηση 8

Να κωδικοποιηθούν σε ψευδογλώσσα:

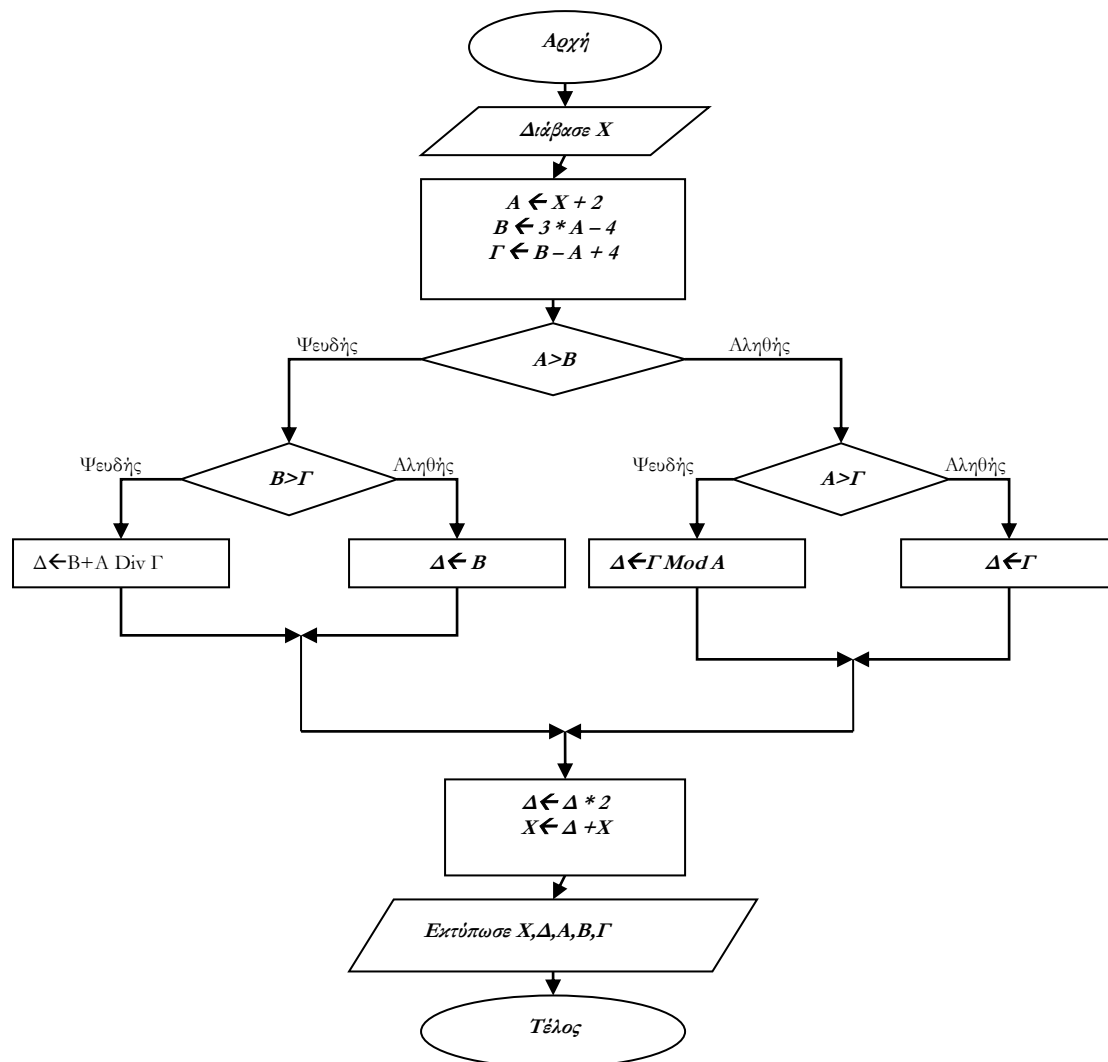
$$1. R = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad 2. E = \frac{1}{2}(B1 + B2).Y \quad 3. K1 = K + \frac{K.X.E}{360}$$

Άσκηση 9

1) Να γράψετε στο τετράδιο σας τις τιμές των μεταβλητών X,A,B,Γ,Δ, όπως αυτές τυπώνονται μετά την εκτέλεση του παρακάτω αλγορίθμου, αν σαν είσοδος δοθεί :

α) X = 1

β) X = -1



2) Να μετατρέψετε το παραπάνω διάγραμμα ροής σε ψευδοκώδικα.

Άσκηση 10 Στο παρακάτω τμήμα του αλγορίθμου να εντοπίσετε το λάθος και να το διορθώσετε. Ποιο από τα αλγοριθμικά κριτήρια δεν πληρείται;

Αλγόριθμος Θέμα2B

Διάβασε X

Διάβασε Y

Αν X <> 0 τότε

$$F \leftarrow (2 * X + 3 * Y) / X + (X - Y) / (Y^2 - 1)$$

$$L \leftarrow F \text{ Div } 2$$

Τέλος_Αν

Εκτύπωσε L

Τέλος Θέμα2B

Άσκηση 11 Ποια από τα παρακάτω αποτελούν εντολές ή εκφράσεις της ψευδογλώσσας:

- 1) $A + B = 10$ 2) $A \leftarrow B * 3$ 3) $A + B \leftarrow 12$ 4) $A \leftarrow 2 * K \leftarrow 32$

Άσκηση 12 Ποια από τα παρακάτω εκχωρεί στη μεταβλητή A τον αριθμό 138

- 1) $A=138$ 2) $A \leftarrow '138'$ 3) $A=: 138$ 4) $A \leftarrow 138$

Άσκηση 13 (Σωστό –Λάθος)

1. Η απόδοση τιμής σε μια μεταβλητή γίνεται μόνο με την εντολή εκχώρησης.
2. Η εντολή ΓΡΑΨΕ Α εμφανίζει το γράμμα Α στην Οθόνη
3. Η εντολή ΓΡΑΨΕ '12*3' Εμφανίζει στην οθόνη 36
4. Η εντολή Met \leftarrow “ ΝΙΚΟΣ” Αποδίδει στη μεταβλητή met την τιμή ΝΙΚΟΣ

Άσκηση 14 Να συμπληρώσετε τον παρακάτω πίνακα:

Εντολή	Εμφάνιση στην οθόνη
ΓΡΑΨΕ 12	
ΓΡΑΨΕ 12+3	
ΓΡΑΨΕ ' 12+3'	
ΦΠΑ \leftarrow 12	
ΓΡΑΨΕ ΦΠΑ	
ΓΡΑΨΕ 12*ΦΠΑ	
ΓΡΑΨΕ '12*ΦΠΑ'	
$A \leftarrow$ ΑΛΗΘΗΣ	
ΓΡΑΨΕ Α	
ΓΡΑΨΕ ' Α'	

Άσκηση 15 Αν $A=4$ και $B= 1$ και $C= 7$ τότε βρείτε ποιες από τις επόμενες λογικές παραστάσεις είναι αληθείς και ποιες ψευδείς.

- 1) $A < B$ ΚΑΙ $B < C$
- 2) $B < C$ ΚΑΙ $(A < B$ Η $A > C)$
- 3) $(B > C$ Η $A > C)$ Η $C > A$
- 4) ΟΧΙ $(A > B)$

Άσκηση 16 Πώς αλλιώς μπορεί να αποδοθεί σε ψευδογλώσσα η παράσταση :
 $x < 0$ ή $x > 10$

- A.** Όχι $x \geq 0$ και Όχι $x \leq 10$ **B.** Όχι $x \geq 0$ ή $x \leq 10$)
Γ. Όχι $(x < 0$ ή $x > 10)$ **Δ.** $x \geq 0$ και Όχι $x \geq 10$

Άσκηση 17 Να γράψετε στο τετράδιο σας τον παρακάτω πίνακα συμπληρωμένο.

Συνθήκη A	Συνθήκη B	Συνθήκη Γ	A ή B και Γ	(A και B) ή Όχι(Γ)	Όχι ((A και B) ή Γ)
Αληθής	Αληθής	Ψευδής			
Αληθής	Ψευδής	Αληθής			
Ψευδής	Αληθής	Αληθής			
Ψευδής	Ψευδής	Αληθής			

Άσκηση 18 Ένας φοιτητής εξετάζεται σε κάποιο μάθημα δύο φορές προφορικά και μία φορά γραπτά. Αν ο γραπτός βαθμός του είναι από πέντε (5) και πάνω τότε υπολογίζεται ο μέσος όρος βαθμολογίας του αθροίζοντας το 30% του μέσου όρου των δύο προφορικών και το 70% του γραπτού του βαθμού. Ανάλογα με το μέσο όρο της βαθμολογίας δίδεται κατάλληλος χαρακτηρισμός με βάση τον παρακάτω πίνακα :

<i>Μέσος Όρος Βαθμολογίας</i>	<i>Χαρακτηρισμός</i>
$5 \leq MO \leq 6.5$	Καλώς
$6.5 < MO < 8.5$	Λιαν Καλώς
$MO \geq 8.5$	Άριστα

Να αναπτύξετε αλγόριθμο ο οποίος :

- α) Να διαβάζει τους δύο προφορικούς και το γραπτό βαθμούς ενός φοιτητή.
- β) Αν ο γραπτός βαθμός του είναι μεγαλύτερος ή ίσος του 5 να υπολογίζει το μέσο όρο βαθμολογίας βάσει του οποίου να τυπώνει τον κατάλληλο χαρακτηρισμό.
- γ) Αν ο γραπτός βαθμός είναι κάτω από 5 τότε να εμφανίζει το μήνυμα "ΑΠΟΤΥΧΩΝ".

Άσκηση 19 Η δημόσια επιχείρηση ηλεκτρισμού υπολογίζει την αξία του λογαριασμού του ηλεκτρικού ρεύματος των καταναλωτών της σύμφωνα με τον παρακάτω πίνακα.

<i>ΗΜΕΡΗΣΙΑ ΚΑΤΑΝΑΛΩΣΗ</i>		<i>ΝΥΧΤΕΡΙΝΗ ΚΑΤΑΝΑΛΩΣΗ</i>	
Κατανάλωση (σε Kwh)	Τιμή / Kwh (σε Ευρώ)	Κατανάλωση (σε Kwh)	Τιμή / Kwh (σε Ευρώ)
Από 1 έως και 800	0.05	Ανεξάρτητα από την κατανάλωση	0.03
Από 801 έως και 1200	0.07		
Από 1201 έως και 1600	0.08		
1601 και πάνω	0.09		

Στην ημερήσια κατανάλωση η χρέωση γίνεται κλιμακωτά ενώ η νυχτερινή δεν εξαρτάται από την κατανάλωση. Στην αξία της ημερήσιας χρέωσης προστίθεται η αξία της νυχτερινής και ένα πάγιο ποσό 15 ευρώ, έτσι ώστε να προκύψει η συνολική αξία του ρεύματος που καταναλώθηκε. Στη συνολική αξία προστίθεται ο φόρος προστιθέμενης αξίας (ΦΠΑ) που είναι 18%, ώστε να προκύψει η τελική αξία του λογαριασμού.

Να κατασκευαστεί αλγόριθμος που :

- α) Να διαβάζει το όνομα, την ημερήσια και τη νυχτερινή κατανάλωση ενός καταναλωτή.
- β) Να υπολογίζει την αξία του ρεύματος που καταναλώθηκε.
- γ) Να υπολογίζει τον ΦΠΑ και την τελική αξία του λογαριασμού.
- δ) Να τυπώνει το όνομα και την συνολική αξία του λογαριασμού του καταναλωτή.

Άσκηση 20 Να γράψετε στο τετράδιο σας τις τιμές των μεταβλητών X,A,B,Γ,Δ, όπως αυτές τυπώνονται μετά την εκτέλεση του παρακάτω αλγορίθμου, αν σαν είσοδος δοθεί :

- α) X=1
- β) X=-1

Αλγόριθμος Θέμα2Α

Διάβασε X

A ← X + 2

B ← 3*A - 4

Γ ← B - A + 4

Αν A > B τότε

 Γ ← Γ - B

 Αν A > Γ τότε

 Δ ← Γ div 2

 Αλλιώς

 Δ ← (Γ+B) mod A

 Τέλος_Αν

X ← X + 1

Αλλιώς

 B ← B div 2 + A

 Γ ← Γ + 2*X

 Αν (B+Γ) mod 3 = 0 τότε

 Δ ← B+Γ

 Αλλιώς

 Δ ← B + A div Γ

 Τέλος_Αν

 X ← X + 3

Τέλος_Αν

Δ ← ((Δ div 2) ^ X) mod 3

X ← Δ + X

Γράψε X, Δ, A, B, Γ

Τέλος Θέμα2B

Άσκηση 21 Για να μπει κάποιος σε κατάσταση με ηλεκτρονικά παιχνίδια πρέπει να ισχύει κάποιο από τα παρακάτω:

- P: Να είναι μεγαλύτερος από 25 χρονών
- Q: Να είναι τουλάχιστον 17 χρονών, αλλά να έχει μαζί του ταυτότητα για να το αποδείξει (T)
- S: Να συνοδεύεται από κάποιον μεγάλο συγγενή του

Να κατασκευάσετε λογική πρόταση (K) ώστε αν η πρόταση κ είναι αληθής θα επιτρέπεται η όχι η πρόσβαση στα ηλεκτρονικά. Η Ηλικία έχει καταχωριστεί στη μεταβλητή ΗΛΙΚΙΑ

Απάντηση :

K ← (ΗΛΙΚΙΑ > 25) Η ((ΗΛΙΚΙΑ >= 17) ΚΑΙ (T = Αληθής)) Η (Σ = αληθής)

3. ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

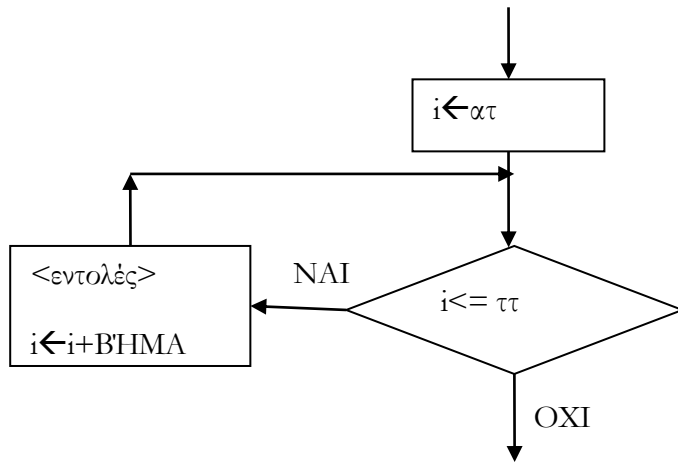
1η Μορφή Επαναληπτικής δομής (ΓΙΑ - ΑΠΟ - ΜΕΧΡΙ)

Οι εντολές επιλογής μας λύνουν αρκετά προβλήματα όχι όμως όλα . Υπάρχουν περιπτώσεις που η διαδικασία εκτέλεσης του αλγορίθμου , απαιτεί την εκτέλεση εντολών που ήδη έχουν εκτελεστεί (επανάληψη) . Πολλές φορές η επανάληψη αυτή απαιτείται να εκτελεστεί περισσότερες από μία φορές. Όταν είναι γνωστός εκ των προτέρων ο αριθμός των επαναλήψεων, μπορεί να χρησιμοποιηθεί η πρώτη μορφή της επαναληπτικής δομής, η οποία σε ψευδοκώδικα έχει την ακόλουθη μορφή:

για <μτ> από <ατ> μέχρι <ττ> με βήμα <μβ>
<ομάδα εντολών>
τέλος_επανάληψης

Διάγραμμα ροής =====>

Εδώ χρησιμοποιούνται οι βοηθητικές λέξεις για, από, μέχρι, με βήμα, τέλος_επανάληψης και οι μεταβλητές-σταθερές μτ (μετρητής), ατ (αρχική τιμή), ττ (τελική τιμή), μβ (μεταβολή).



Αυτή η μορφή της επαναληπτικής δομής έχει το χαρακτηριστικό ότι απαγορεύεται στην ομάδα εντολών να υπάρχουν εντολές μεταβολής της τιμής του μετρητή, η οποία μεταβάλλεται από την ίδια την εντολή σε κάθε επανάληψη κατά <μεταβολή>. Συνήθως, όταν η μεταβολή είναι 1, εννοείται και δεν αναφέρεται καθόλου.

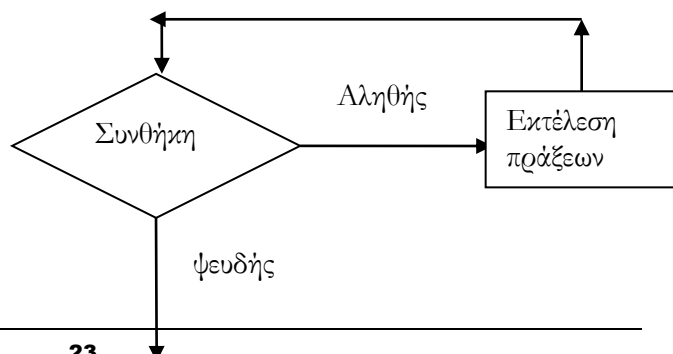
2η Μορφή Επαναληπτικής Δομής (ΟΣΟ - ΕΠΑΝΑΛΑΒΕ)

Η δεύτερη μορφή της επαναληπτικής δομής σε ψευδοκώδικα έχει την ακόλουθη μορφή:

όσο <συνθήκη> επανάλαβε
<ομάδα εντολών>
τέλος_επανάληψης

Διάγραμμα ροής =====>

Εδώ χρησιμοποιούνται οι βοηθητικές λέξεις όσο, επανάλαβε, τέλος_επανάληψης. Μετά τη



βοηθητική λέξη **όσο** ακολουθεί η συνθήκη συνέχειας. Αμέσως μετά, μεταξύ των λέξεων **επανάλαβε** και **τέλος_επανάληψης** γράφεται η ομάδα εντολών. Οι συνθήκες που χρησιμοποιούνται εδώ είναι ίδιες με τις συνθήκες που χρησιμοποιούνται στην εντολή επιλογής.

Στην αρχή της εκτέλεσης ελέγχεται η συνθήκη. Αν αυτή ικανοποιείται, τότε εκτελείται η ομάδα εντολών. Στη συνέχεια, ελέγχεται ξανά η συνθήκη. Δεν είναι σίγουρο ότι αυτή εξακολουθεί να είναι αληθής διότι το αποτέλεσμα της μπορεί να έχει επηρεαστεί από την εκτέλεση της ομάδας εντολών. Αν η συνθήκη ικανοποιείται ξανά, εκτελείται πάλι η ομάδα εντολών. Η διαδικασία συνεχίζεται, όσο η συνθήκη παραμένει αληθής. Όταν η συνθήκη συνέχειας γίνει ψευδής, η ομάδα εντολών δεν εκτελείται και τότε λήγει η εκτέλεση της δομής.

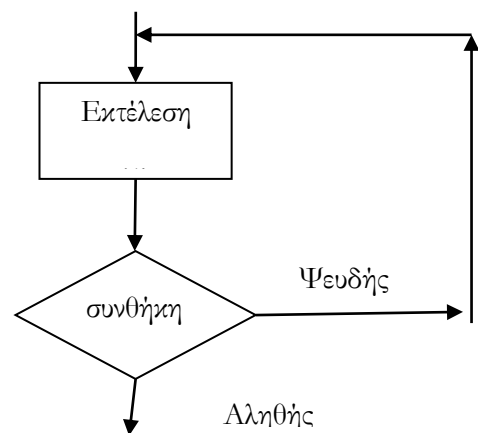
Αυτή η μορφή της επαναληπτικής δομής έχει το χαρακτηριστικό **ότι υποχρεωτικά στην ομάδα εντολών πρέπει να υπάρχουν εντολές μεταβολής της συνθήκης.** Επίσης επειδή η συνθήκη βρίσκεται στην πρώτη εντολή η εκτέλεση της εντολής μπορεί να μη γίνει καθόλου (Αν η συνθήκη είναι ψευδής)

3η Μορφή Επαναληπτικής δομής (ΜΕΧΡΙΣ_ΟΤΟΥ)

Η τρίτη μορφή της επαναληπτικής δομής σε ψευδοκώδικα έχει την ακόλουθη μορφή:

```

αρχή_επανάληψης
    <ομάδα εντολών>
μέχρις_ότου <συνθήκη τέλους>
    
```



Εδώ χρησιμοποιούνται οι βοηθητικές λέξεις **αρχή_επανάληψης**, **μέχρις_ότου**. Μετά τη βοηθητική λέξη **αρχή_επανάληψης** ακολουθούν οι εντολές οι οποίες τελειώνουν πριν από τη λέξη **μέχρις_ότου**. Και στη περίπτωση αυτή, οι συνθήκες που χρησιμοποιούνται είναι ίδιες με τις συνθήκες που χρησιμοποιούνται στην εντολή επιλογής. Αξίζει να σημειωθεί ότι επειδή η εντολή τελειώνει με το **μέχρις_ότου** <συνθήκη τέλους>, δε χρειάζεται καμία δεσμευμένη λέξη τέλους εντολής.

Σε αυτή τη μορφή της εντολής επανάληψης **ο έλεγχος γίνεται μετά την εκτέλεση της ομάδας εντολών. Αυτό σημαίνει ουσιαστικά ότι η ομάδα εντολών εκτελείται τουλάχιστον μία φορά.** Μετά την εκτέλεση της ομάδας εντολών ελέγχεται η συνθήκη. Αν αυτή δεν ικανοποιείται, τότε εκτελείται ξανά η ομάδα εντολών, όσο η συνθήκη παραμένει ψευδής. Όταν η συνθήκη γίνει αληθής, λήγει η εκτέλεση της επαναληπτικής δομής

ΠΑΡΑΤΗΡΗΣΕΙΣ

☼ Η πρώτη μορφή επαναληπτικής διαδικασίας χρησιμοποιείται **όταν γνωρίζουμε πόσες φορές θα γίνει η επανάληψη.**

☼ Στη δεύτερη μορφή επανάληψης **δεν γνωρίζουμε το πλήθος των επαναλήψεων** (Αυτό καθορίζεται από τη συνθήκη) **.Δηλαδή πριν μπούμε στην επανάληψη είναι απαραίτητο να ελέγξουμε τη συνθήκη, όπως επίσης στο τέλος των εντολών μέσα στο βρόχο πρέπει να ελεγχθεί επίσης η συνθήκη.** Διαφορετικά υπάρχει κίνδυνος να μπούμε σε ένα βρόχο χωρίς τέλος (ατέρμων βρόχος)

Έτσι το πλεονέκτημα αυτής της μορφής είναι ότι η συνθήκη είναι πλήρως ελεγχόμενη που σημαίνει ότι αν θέλουμε δεν εκτελείται καθόλου η επανάληψη.

☼ Η Τρίτη μορφή επανάληψης έχει το χαρακτηριστικό ότι η **συνθήκη ελέγχεται στο τέλος του βρόχου**. Άρα θα πρέπει να υπάρχει μέσα στο βρόχο εντολή τροποποίησης της συνθήκης. Σε αυτή τη μορφή οι εντολές που βρίσκονται μέσα στο βρόχο **θα εκτελεστούν τουλάχιστον μία φορά**. Αυτό μπορεί να θεωρηθεί μειονέκτημα αλλά θα δείτε από τα παραδείγματα ότι αυτό είναι απαραίτητο πολλές φορές.

☼ Στην πρώτη μορφή επανάληψης **για να λειτουργήσει ο μετρητής (μτ) θα πρέπει η αρχική τιμή (ατ) να είναι μικρότερη της τελικής (ττ) αν το βήμα είναι θετικό**. Ενώ αν η αρχική τιμή (ατ) είναι μεγαλύτερη της τελικής (ττ) τότε το βήμα πρέπει να είναι αρνητικό, Επίσης οι τιμές από , μέχρι και βήμα δεν είναι ανάγκη να είναι ακέραιες. Μπορούν να λάβουν οποιαδήποτε πραγματική τιμή.

Έτσι

Ο βρόχος Για n από 5 μέχρι 5 εκτελείται ακριβώς μία φορά

Για n από 5 μέχρι 1 δεν εκτελείται καμία φορά

Για n από 100 μέχρι 1 με_βήμα -1 εκτελείται κανονικά

Για n από 0 μέχρι 1 με_βήμα 0,001 εκτελείται κανονικά



ΑΣΚΗΣΕΙΣ ΣΤΙΣ ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΔΟΜΕΣ

Άσκηση 1 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει το μέσο όρο, το άθροισμα και το γινόμενο των αριθμών από το 1 έως το N .

Άσκηση 2 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει το μέσο όρο, το άθροισμα και το γινόμενο των περιττών αριθμών από το 1 έως το N

Άσκηση 3 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει το μέσο όρο, το άθροισμα και το γινόμενο των άρτιων αριθμών από το 1 έως το N .

Άσκηση 4 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει το άθροισμα της πιο κάτω σειράς: $\Sigma = 1^n + 2^n + 3^n + \dots + 100^n$

Άσκηση 5 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει το γινόμενο της πιο κάτω σειράς: $\Pi = 1^n * 2^n * 3^n * \dots * 10^n$

Άσκηση 6 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει το αποτέλεσμα

των πιο κάτω παραστάσεων: $\sum_{n=1}^k n(2n-1)^3$ και $\sum_{n=1}^{10} n^2 2^n$ (Να γίνει μετά το αντίστοιχο μάθημα Στατιστικής)

Άσκηση 7 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει τους 30 πρώτους όρους των πιο κάτω σειρών και να τυπώνει το αποτέλεσμά τους:

$$s1 = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots \quad \text{και}$$

$$s2 = 1 + x + \frac{x^2}{2^2} + \frac{x^3}{3^2} + \frac{x^4}{4^2} + \dots$$

Άσκηση 8 Να γραφεί αλγόριθμος ο οποίος για ένα σύνολο N τυχαίων ακεραίων αριθμών που δίδονται από το πληκτρολόγιο να υπολογίζει και να τυπώνει (α) το μέσο όρο τους και (β) πόσοι από αυτούς είναι θετικοί, πόσοι αρνητικοί και πόσοι μηδέν

Άσκηση 9 Να γραφεί αλγόριθμος ο οποίος για ένα σύνολο N τυχαίων ακεραίων αριθμών που δίδονται από το πληκτρολόγιο να υπολογίζει και να τυπώνει (α) το μέσο όρο τους και (β) πόσοι από αυτούς είναι άρτιοι και πόσοι περιττοί.

Άσκηση 10 Να γραφεί αλγόριθμος ο οποίος να υπολογίζει και να τυπώνει την τιμή της πιο κάτω σειράς:

$$S1 = 1! + 2! + 3! + \dots + N!$$

Σημείωση: Το σύμβολο ! είναι το παραγοντικό

(π.χ. $2! = 1 * 2$, $3! = 1 * 2 * 3$, $4! = 1 * 2 * 3 * 4$, κ.ο.κ.)

ΜΕΤΑΤΡΟΠΕΣ ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ

1. Μετατροπή της δομής **όσο** σε **μέχρις_ότου**

Εντολή Όσο.....	Εντολή Μέχρις_ότου
<p>Όσο συνθήκη επανέλαβε Ομάδα εντολών Τέλος_επανάληψης</p>	<p>Αν συνθήκη τότε αρχή_επανάληψης ομάδα εντολών μέχρις_ότου ΟΧΙ(συνθήκη) Τέλος_αν</p>

☼ Η συνθήκη που υπάρχει στην Όσο μετατρέπεται σε ΟΧΙ(συνθήκη) στην μέχρις_ότου

☼ Επειδή η εντολή όσο μπορεί να μην εκτελεστεί καθόλου, γιαυτό απαιτείται η μέχρις_ότου να μπει μέσα στην επιλογή αν. Τότε για να εξασφαλίσουμε ότι, όταν απαιτείται δεν θα εκτελεστεί.

2. Μετατροπή της δομής **μέχρις_ότου** σε **όσο.....**

Εντολή Μέχρις_ότου	Εντολή Όσο.....
<p>αρχή_επανάληψης ομάδα εντολών μέχρις_ότου Συνθήκη</p>	<p>Ομάδα εντολών Όσο ΟΧΙ(συνθήκη) επανέλαβε Ομάδα εντολών Τέλος_επανάληψης</p>

☼ Επειδή η μέχρις_ότου εκτελείται τουλάχιστον μία φορά, όταν γίνεται η μετατροπή, θα πρέπει οι εντολές που βρίσκονται στο σώμα της, να προηγηθούν της όσο γιατί η όσο, ενδέχεται να μην εκτελεστεί ποτέ.

☼ Η συνθήκη στην μέχρις_ότου μετατρέπεται στην όσο, σε ΟΧΙ(συνθήκη)

ΓΕΝΙΚΗ ΠΑΡΑΤΗΡΗΣΗ

1. Στην επαναληπτική δομή **Όσο...επανάλαβε**, η συνθήκη ονομάζεται **συνθήκη συνέχειας**, ενώ στην επαναληπτική δομή **Αρχή_επανάληψης...μέχρις_ότου**, η συνθήκη ονομάζεται **συνθήκη τερματισμού**.
2. Η συνθήκη συνέχειας P και Q, μετά τη μετατροπή της επαναληπτικής δομής, πρέπει να γίνει P ή Q. Δηλαδή αντιστρέφονται οι λογικοί τελεστές, το λογικό και γίνεται λογικό ή και αντίστροφα

- 3 Η συνθήκη συνέχειας $P > Q$, μετά τη μετατροπή της επαναληπτικής δομής, πρέπει να γίνει $P \leq Q$. Δηλαδή μπαίνει ο συμπληρωματικός τελεστής συσχέτισης
- 4 Η συνθήκη τερματισμού μπορεί να προκύψει πολύ απλά από τη συνθήκη συνέχειας ως εξής: συνθήκη τερματισμού = Όχι (συνθήκη συνέχειας) και αντίστροφα .

3. Μετατροπή της δομής γιασε Όσο.....

Η δομή Για μετατρέπεται πάντα σε όσο..... και στη συνέχεια σεΜέχρις_ότου . Εδώ θα πρέπει τις λειτουργίες που εκτελεί η Για αυτόματα , θα πρέπει να τις ορίσουμε εμείς.

Εντολή Για.....	Εντολή Όσο....
<p>Για j από ατ μέχρι ττ με βήμα β ομάδα εντολών τέλος_επανάληψης</p>	<p>J ← ατ Όσο j ≤ ττ επανέλαβε Ομάδα εντολών j ← j+β Τέλος_επανάληψης</p>

4. Οι δομές μέχρις_ότου και όσο..... δεν μετατρέπονται πάντα σε Για.....

Για να μπορεί να γίνει η μετατροπή πρέπει να γνωρίζουμε τον

- αριθμό των επαναλήψεων
- την αρχική τιμή της μεταβλητής επανάληψης.
- Την τελική τιμή της μεταβλητής επανάληψης
- Να μπορεί να προσδιοριστεί το βήμα και να είναι σταθερό

Η μετατροπή γίνεται με τον παρακάτω τρόπο :

Εντολή Όσο.....	Εντολή Για.....
<p>Όσο μτε Τελεστής σύγκρισης επανέλαβε Ομάδα εντολών μτε ← μτε+β Τέλος_επανάληψης</p>	<p>Για μτε από ατ μέχρι ττ με_βήμα β ομάδα εντολών Τέλος_επανάληψης</p>



ΑΣΚΗΣΕΙΣ στις ΜΕΤΑΤΡΟΠΕΣ αλγορίθμων

Άσκηση 1. Δίνεται ο παρακάτω αλγόριθμος:

Αλγόριθμος Πολλαπλή

Διάβασε α

Επίλεξε α

Περίπτωση < 0

εμφάνισε “Αρνητικό”

Περίπτωση ≤ 100

εμφάνισε “Μεταξύ 0 και 100”

Περίπτωση ≤ 200

εμφάνισε “Μεταξύ 100 και 200”

Περίπτωση αλλιώς

εμφάνισε “Μεγαλύτερο από 200”

Τέλος_επιλογών

Τέλος Πολλαπλή

Να αναπτύξετε αλγόριθμο ισοδύναμο με τον παραπάνω, χρησιμοποιώντας εμφωλευμένες επιλογές.

Άσκηση 2 Να γραφούν τα παρακάτω τμήματα αλγορίθμου χρησιμοποιώντας την δομή για...από...μέχρι:

1)

Διάβασε X, Σ

Αρχή_επανάληψης

$\Sigma \leftarrow X * \Sigma$

$X \leftarrow X - 1$

Μέχρις_ότου $X < 5$

2)

$X \leftarrow 3$

Διάβασε Σ

Αρχή_επανάληψης

$\Sigma \leftarrow X * \Sigma$

$X \leftarrow X + 1$

Μέχρις_ότου $X > 5$

Δομές δεδομένων και Αλγόριθμοι

Μέρος

2

Περιεχόμενα

- 1. Δεδομένα – Δομές δεδομένων**
- 2. Δυναμικές-Στατικές δομές δεδομένων**
- 3. 1^η Μορφή δομής δεδομένων ΠΙΝΑΚΕΣ(table)**
- 4. 2^η Μορφή δομής δεδομένων ΕΓΓΡΑΦΗ(Record)**
- 4. 3^η Μορφή δομής Δεδομένων ΣΤΟΙΒΑ (Stack)**
- 5. 4^η Μορφή δομής δεδομένων ΟΥΡΑ (Queue)**

1. Δεδομένα -Δομές Δεδομένων

Ας υποθέσουμε ότι θέλουμε να υπολογίσουμε τον μισθό ενός υπαλλήλου. Αυτός εξαρτάται από το **βασικό μισθό** ,τα **χρόνια υπηρεσίας** που έχει από το **χρονοεπίδομα** που παίρνει ,από τον **αριθμό των παιδιών** του κλπ. Βλέπουμε ότι απαιτούνται κάποια **στοιχεία (δεδομένα)** πάνω στα οποία θα γίνει κάποια **επεξεργασία** για να υπολογιστεί ο μισθός του υπαλλήλου.

Με τον όρο **Δεδομένα (data)** θα εννοούμε κάθε παράσταση γεγονότων ή εννοιών που είναι κατάλληλη για επικοινωνία ή επεξεργασία από τον άνθρωπο ή αυτόματα μέσα (H/Y-Μηχανές)



Μπορεί να θεωρηθεί ότι ο αλγόριθμος είναι το μέσο που θα επεξεργαστεί τα δεδομένα για να παραχθούν τα αποτελέσματα (Πληροφορία) . Τα δεδομένα ενός προβλήματος αποθηκεύονται στον H/Y είτε στην Κύρια μνήμη (RAM) είτε σε αποθηκευτικά μέσα (περιφερειακές μονάδες H/Y) . Η αποθήκευση αυτή δεν γίνεται κατά ένα τυχαίο τρόπο αλλά συστηματικά , δηλαδή για την αποθήκευση των δεδομένων χρησιμοποιούμε μια **δομή** . Έτσι λοιπόν μπορούμε να πούμε ότι :

Δομή δεδομένων (data Structure) είναι ένα σύνολο αποθηκευμένων δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών (πράξεων) πάνω σε αυτά

Η σχέση μεταξύ αλγορίθμων και δομών δεδομένων είναι ιδιαίτερα στενή. Ο Witth , δημιουργός της γλώσσας προγραμματισμού PASCAL είχε διατυπώσει τη χαρακτηριστική εξίσωση :

Αλγόριθμοι +Δομές δεδομένων =Προγράμματα

Μερικές βασικές λειτουργίες (πράξεις) πάνω στις δομές δεδομένων είναι:

Προσπέλαση (access) Πρόσβαση σε κάποιο δεδομένο με σκοπό να εξεταστεί ή να τροποποιηθεί

Εισαγωγή (Insertion) προσθήκη δεδομένων μέσα στην υπάρχουσα δομή

Διαγραφή (Deletion) Αφαίρεση δεδομένων μέσα από μια υπάρχουσα δομή

Αναζήτηση (searching) προσπέλαση των δεδομένων μέχρι να βρεθεί το αναζητούμενο δεδομένο που έχει κάποια χαρακτηριστική ιδιότητα , πάνω στην οποία βασίζεται η αναζήτηση.

Ταξινόμηση (Sorting) Διάταξη των δεδομένων κατά αύξουσα ή φθίνουσα σειρά ως προς κάποιο χαρακτηριστικό τους.

Αντιγραφή (Copying) πράξη που μας δίνει τη δυνατότητα να αντιγράψουμε ένα ή περισσότερα δεδομένα σε μια άλλη δομή

Συγχώνευση (merging) Ενέργεια (πράξη) κατά την οποία δύο ή περισσότερες δομές μπορούν να συνενωθούν σε μία νέα δομή

Διαχωρισμός (separation) Η αντίστροφη πράξη της συγχώνευσης , δηλαδή μία δομή να διασπάται σε δύο ή περισσότερες δομές .

ΠΑΡΑΤΗΡΗΣΗ

Δεν χρησιμοποιούνται όλες οι παραπάνω πράξεις (οικτώ) σε όλες τις δομές δεδομένων. Υπάρχουν δομές που είναι αποδοτικότερες σε κάποια από τις παραπάνω πράξεις , ενώ άλλες δομές είναι λιγότερο αποδοτικές . Αυτός είναι και ο λόγος που υπάρχουν πολλές δομές δεδομένων . Ανάλογα με τη εργασία που θέλουμε να κάνουμε επιλέγουμε να χρησιμοποιήσουμε και την κατάλληλη δομή.

2. Δυναμικές-Στατικές δομές δεδομένων

Οι δομές δεδομένων χωρίζονται σε δύο μεγάλες κατηγορίες :

A. Στατικές δομές δεδομένων (static)

B. Δυναμικές δομές δεδομένων (dynamic)

Στην πρώτη κατηγορία υπάγονται , **οι Πίνακες (array) , Εγγραφές (Records)** και άλλες ενώ στη δεύτερη τα **αρχεία (files) ,Στοιβες (stacks) ,Ουρές (Queues)** κλπ

Στις στατικές δομές δεδομένων , το μέγεθος τους είναι προκαθορισμένο από τη στιγμή της κατασκευής του προγράμματος και δεν είναι δυνατόν να αλλάξει κατά την εκτέλεση του προγράμματος. Τα στοιχεία μιας στατικής δομής δεδομένων αποθηκεύονται σε συνεχόμενες θέσεις της κύριας μνήμης.(RAM)

Στις δυναμικές δομές δεδομένων ,το μέγεθός τους δεν είναι προκαθορισμένο από την αρχή της κατασκευής του προγράμματος , αλλά προσδιορίζεται όταν εκτελείται το πρόγραμμα ανάλογα με τις ανάγκες του προβλήματος. Τα στοιχεία μιας δυναμικής δομής δεδομένων συνήθως αποθηκεύονται σε βοηθητικές μνήμες και όχι σε συνεχόμενες θέσεις της κύριας μνήμης (RAM)

3. 1^η Μορφή Δομής δεδομένων ΠΙΝΑΚΕΣ

Είναι χρήσιμο πολλές φορές να μπορούμε να ταξινομήσουμε διάφορες πληροφορίες , με τέτοιο τρόπο ώστε να μπορούμε να αναφερθούμε σ' αυτές χρησιμοποιώντας τη θέση που βρίσκονται . Ας υποθέσουμε για παράδειγμα ότι θέλουμε να παραστήσουμε το δυναμικό ενός σχολείου κατά φύλο και τάξη, θα μπορούσε να παρασταθεί όπως παρακάτω

Τάξεις	A	B	Γ
---------------	----------	----------	----------

Αγόρια	122	85	72
Κορίτσια	111	82	54

Έτσι καταλαβαίνουμε ότι τα κορίτσια της Β τάξεως είναι 82 , τα αγόρια της Γ' τάξεως είναι 72 κ.ο.κ. Αν παραλείψουμε τις κεφαλίδες (πρώτη γραμμή πρώτη στήλη) θα προκύψει η παρακάτω ορθογώνια διάταξη αριθμών που θα λέγεται

122	85	72
111	82	54

Πίνακας (Array) . Ο πίνακας αυτός θα λέμε ότι έχει δύο γραμμές και τρεις στήλες (2x3) . Για να αναφερθούμε σε ένα στοιχείο ενός πίνακα χρησιμοποιούμε το όνομα του πίνακα και μέσα σε αγκύλες δύο δείκτες που ο πρώτος μας δείχνει τη γραμμή και ο δεύτερος τη στήλη . Για παράδειγμα αν το όνομα του πίνακα είναι Σχολείο για να αναφερθούμε στα αγόρια της Β ' τάξεως γράφουμε σχολείο[1,2] ή στα κορίτσια της Γ' τάξεως γράφουμε σχολείο[2,3]

Δείτε ένα άλλο παράδειγμα πίνακα :

Ας υποθέσουμε ότι θέλουμε να τοποθετήσουμε σ'ένα πίνακα την πρόοδο 10 μαθητών ενός τμήματος . Έτσι θα έχω τον πίνακα

Bathmoi=[12,18,16,9,17,20,19,12,13,16]

Τώρα γνωρίζω ότι Bathmoi[3]=16 , που σημαίνει ότι ο τρίτος μαθητής έχει βαθμό 16. Εδώ παρατηρώ ότι για να αναφερθώ σε ένα στοιχείο του πίνακα χρησιμοποιώ ένα δείκτη αφού ο πίνακας μου είναι μιας γραμμής (ή μιας στήλης)

Στη πληροφορική αντιμετωπίζουμε τους πίνακες **σαν μεταβλητές με δείκτες**. Οι δείκτες αυτοί προσδιορίζουν τη θέση του στοιχείου μέσα στον πίνακα.

Με το όνομα του πίνακα περιγράφουμε ένα σύνολο από τιμές

Αν ο πίνακας έχει μια γραμμή (ή μια στήλη) απαιτείται ένας δείκτης για τον προσδιορισμό της θέσης του στοιχείου, ονομάζεται πίνακας **μιας διάστασης**

Αν ο πίνακας έχει γραμμές και στήλες απαιτούνται δύο δείκτες για τον προσδιορισμό της θέσης ενός στοιχείου. Υποχρεωτικά ο πρώτος προσδιορίζει τη γραμμή και ο δεύτερος τη στήλη. Ο πίνακας αυτός ονομάζεται πίνακας **δύο διαστάσεων**

Μπορούμε να κατασκευάσουμε πίνακα **τριών διαστάσεων** Για παράδειγμα φανταστείτε μια καρτέλα γιμάτη με τις καρτέλες προόδου των μαθητών του τμήματός σας . Η κάθε καρτέλα είναι ένας πίνακας δύο διαστάσεων (γραμμές και στήλες) και όλες μαζί είναι ένας πίνακας τριών διαστάσεων. Για να αναφερθούμε σ'ένα στοιχείο ενός πίνακα τριών διαστάσεων πρέπει να χρησιμοποιήσουμε τρεις δείκτες . Για το παραπάνω παράδειγμα Karteles[i,j,k] . Ο Πρώτος δείκτης μας δείχνει τη γραμμή , ο δεύτερος τη στήλη και ο τρίτος τη καρτέλα.

Ένας πίνακας που έχει δύο διαστάσεις ίσες θα ονομάζεται **τετραγωνικός** πίνακας..

☼ Στη πράξη οι **στατικές δομές** δεδομένων υλοποιούνται κυρίως με τη δομή του πίνακα που υποστηρίζονται από όλες τις γλώσσες προγραμματισμού .

Το μέγεθος ενός πίνακα είναι προκαθορισμένο από τη στιγμή του προγραμματισμού και όχι κατά την εκτέλεση του προγράμματος.

Τα στοιχεία της δομής του πίνακα αποθηκεύονται σε συνεχόμενες θέσεις της βοηθητικής μνήμης.

Κατά την εκτέλεση ενός αλγόριθμου επεξεργαζόμαστε συνήθως ένα στοιχείο του πίνακα κάθε φορά. Στις εντολές λοιπόν των αλγορίθμων εμφανίζονται, εν γένει, τα στοιχεία του πίνακα και όχι όλος ο πίνακας

ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ

Καταχώριση στοιχείων μονοδιάστατου πίνακα στη μνήμη Η/Υ

Αν Θέλουμε να καταχωρίσουμε στη μνήμη του Η/Υ τα εισιτήρια που κόπηκαν σε θέατρο για ένα μήνα , μπορούμε να χρησιμοποιήσουμε 30 μεταβλητές και να ειχωρήσουμε την κάθε μία μεταβλητή τα εισιτήρια μίας μέρας. Αυτό όμως στους αλγορίθμους με τους πίνακες μπορεί να γίνει με τον παρακάτω τρόπο:

Για παράδειγμα έστω ότι ο πίνακας των εισιτηρίων

$EIS = [205, 197, 89, 231, 148, 150, 176, 122, 134, 145]$ μπορεί να καταχωριστεί στη μνήμη του Η/Υ δίνοντας τα στοιχεία από το πληκτρολόγιο με το παρακάτω τμήμα αλγορίθμου :

Για x από 1 μέχρι 30

 Διάβασε $EIS[x]$

Τέλος_επανάληψης

Ανάγνωση και εκτύπωση στοιχείων από τη μνήμη του Η/Υ

Αν θέλουμε να εκτυπώσουμε τα στοιχεία του παραπάνω πίνακα μπορούμε να γράψουμε το παρακάτω τμήμα αλγορίθμου :

Για x από 1 μέχρι 30

 τύπωσε $EIS[x]$

Τέλος_επανάληψης

Καταχώριση στοιχείων διδιάστατου πίνακα στη μνήμη του Η/Υ

Ο παρακάτω πίνακας περιέχει τους βαθμούς τριών μαθητών .

	Α ΤΡΙΜ	Β ΤΡΙΜ	Γ ΤΡΙΜ	ΓΡΑΠΤΑ
Μαθητής 1	12	16	17	15
Μαθητής 2	16	17	12	14
Μαθητής 3	11	15	16	17

Αν θέλουμε να καταχωρίσουμε στη μνήμη του Η/Υ τους βαθμούς αυτούς δίνοντας τα δεδομένα από το πληκτρολόγιο αρκεί να γράψουμε το παρακάτω τμήμα αλγορίθμου:

για I από 1 μέχρι 3

 Για j από 1 μέχρι 4

 Διάβασε $B[I,j]$

 Τέλος_Επανάληψης

Τέλος_Επανάληψης

Ανάγνωση και εκτύπωση στοιχείων από τη μνήμη του Η/Υ

για I από 1 μέχρι 3

Για j από 1 μέχρι 4

τύπωσε B[I,j]

Τέλος_Επανάληψης

Τέλος_Επανάληψης

ΠΑΡΑΤΗΡΗΣΕΙΣ

☀ Δύο Πίνακες A και B που έχουν την ίδια διάσταση μην θα λέμε ότι είναι ίσοι αν και μόνο αν τα αντίστοιχα στοιχεία τους ταυτίζονται. Για παράδειγμα $A[i,j]=B[i,j]$ για κάθε $i=1,2,3,4,5, \dots, m$ και $j=1,2,3,4, \dots, n$

☀ Η τιμή κάθε στοιχείου του πίνακα αποθηκεύεται σένα τμήμα της μνήμης του Η/Υ, που δεσμεύεται ειδικά γι' αυτό. Οι τιμές των στοιχείων του πίνακα καταχωρούνται σε γειτονικές θέσεις στη μνήμη και δεν είναι δυνατόν να υπερβούμε τα όρια της δήλωσης του πίνακα. Δηλαδή το μέγεθος του πίνακα προορίζεται στην αρχή του αλγορίθμου.

☀ Το είδος των στοιχείων ενός πίνακα είναι καθορισμένο από την αρχή. Αυτό σημαίνει ότι ένας πίνακας περιέχει ενός μόνο είδους στοιχεία, που μπορεί να είναι Ακέραιοι, πραγματικοί, αλφαριθμητικά (string), λογικές τιμές (Αληθής ή ψευδής)

☀ Συνηθίζεται αφού το μέγεθος είναι προκαθορισμένο, συνηθίζεται στη διαχείριση ενός πίνακα να χρησιμοποιούμε την επαναληπτική δομή Για τέλος_επανάληψης

☀ Έτσι μπορούμε να πούμε ότι οι πίνακες έχουν τα παρακάτω μειονεκτήματα και πλεονεκτήματα

Μειονεκτήματα

Σταθερό μέγεθος, άρα σχετική δυσκολία στον προγραμματισμό
Δέσμευση μνήμης Η/Υ καθ' όλη τη διάρκεια εκτέλεσης του προγράμματος. Άρα χρησιμοποιούνται σε μικρό πλήθος δεδομένων

Πλεονεκτήματα

Σχετικά εύκολη χρήση των πινάκων

Με το ίδιο όνομα αναφερόμαστε σε όλα τα στοιχεία του πίνακα, χρησιμοποιώντας δείκτες αυτό σημαίνει ότι δεν χάνονται τα δεδομένα κατά την εκτέλεση του προγράμματος.

ΤΥΠΙΚΕΣ ΕΠΕΞΕΡΓΑΣΙΕΣ ΠΙΝΑΚΩΝ

Υπολογισμός αθροισμάτων στοιχείων πίνακα

Για μονοδιάστατο πίνακα

sum ← 0

Για χ από 1 μέχρι N

sum ← sum + pin[x]

τέλος_επανάληψης

Για πίνακα δύο διαστάσεων

sum ← 0

Για i από 1 μέχρι N

Για j από 1 μέχρι M

```
sum ← sum + rin[I,j]
τέλος_επανάληψης
τέλος_επανάληψης
```

Εύρεση του μέγιστου και του ελάχιστου στοιχείων πίνακα

Παράδειγμα 1 Εύρεση του μικρότερου στοιχείου μονοδιάστατου πίνακα
Δίνεται ένας μονοδιάστατος πίνακας table[100]. Να σχεδιαστεί αλγόριθμος που να βρίσκει το μικρότερο στοιχείο του.

```
Αλγόριθμος Min
Δεδομένα // table //
min ← table[1]
Για I από 2 μέχρι 100
  Αν table[i] < min τότε
    min ← table[i]
  τέλος_αν
τέλος_επανάληψης
Αποτελέσματα // min //
Τέλος Min
```

Παράδειγμα 2 Εύρεση του μικρότερου στοιχείου διδιάστατου πίνακα

```
Αλγόριθμος Min2
Δεδομένα // Table //
min ← table[1]
Για i από 2 μέχρι N
  Για j από 1 μέχρι M
    Αν table[i,j] < min τότε
      min ← table[i,j]
  Τέλος_αν
τέλος_επανάληψης
τέλος_επανάληψης
Αποτελέσματα // min //
Τέλος Min2
```

Θα μπορούσαμε να κρατήσουμε τη θέση που βρίσκεται το μικρότερο στοιχείο του πίνακα σε μια μεταβλητή Position έτσι ο παραπάνω αλγόριθμος θα γινόταν :

```
Αλγόριθμος Min_Θέση
Δεδομένα // table //
min ← table[1]
position ← 1
Για I από 2 μέχρι 100
  Αν table[i] < min τότε
    min ← table[i]
    position ← i
  τέλος_αν
```

```
τέλος_επανάληψης
Αποτελέσματα // min , position//
Τέλος Min_Θέση
```

Με την ίδια λογική θα μπορούσαμε να κρατούσαμε τη θέση του μικρότερου στοιχείου ενός διδιάστατου πίνακα . Θα χρειαζόμασταν όμως δύο μεταβλητές μία μεταβλητή για να κρατήσουμε τη γραμμή (Gr) και μία μεταβλητή για να κρατήσουμε τη στήλη (St). Έτσι θα είχαμε τον παρακάτω αλγόριθμο :

```
Αλγόριθμος Min2_Θέση
Δεδομένα // Table//
min ← table[1,1]
Gr ← 1
St ← 1
Για i από 2 μέχρι N
    Για j από 1 μέχρι M
        Αν table[i,j]<min τότε
            min ← table[i,j]
            Gr ← i
            St ← j
        Τέλος_αν
    τέλος_επανάληψης
τέλος_επανάληψης
Αποτελέσματα // min ,Gr , St//
Τέλος Min2
```

Αναζήτηση στοιχείων πίνακα

Η Αναζήτηση στοιχείου σε κάποιο πίνακα είναι ένα θέμα ιδιαίτερα ενδιαφέρον αφού σε πάρα πολλές εφαρμογές ,απαιτείται να ψάξουμε να βρούμε μια πληροφορία σένα πίνακα. Υπάρχουν πολλές μέθοδοι αναζήτησης (searching) που οι διαφορές οφείλονται στο αν είναι ταξινομημένος ο πίνακας ή όχι.

Η πιο απλή μορφή αναζήτησης σένα πίνακα είναι η σειριακή ή γραμμική μέθοδος που θα περιγράψουμε παρακάτω : Με αυτή τη μέθοδο ψάχνουμε το αναζητούμενο στοιχείο σειριακά –ακολουθιακά ξεκινώντας από το πρώτο στοιχείο του πίνακα. Ο αλγόριθμος συγκρίνει κάθε στοιχείο του πίνακα με την αναζητούμενη τιμή. Έτσι αν εντοπιστεί το αναζητούμενο στοιχείο , η μέθοδος σταματά διαφορετικά ο αλγόριθμος μας δίνει κατάλληλο μήνυμα.

Αυτή η μέθοδος είναι αρκετά χρονοβόρα και εφαρμόζεται σε πίνακες μη ταξινομημένους και με μικρό μέγεθος γενικά.

1. Σειριακή αναζήτηση σε μη ταξινομημένο πίνακα , κάθε στοιχείο του πίνακα είναι μοναδικό.(μία φορά)

```
Αλγόριθμος Σειριακή_1
Δεδομένα // N, A, key//
done ← ψευδής
```

```

position ← 0
i ← 1
Όσο (done = ψευδής) και (i ≤ N) επανάλαβε
    Αν A[i] = key τότε
        done ← αληθής
        position ← i
    αλλιώς
        i ← i + 1
Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // done, position //
Τέλος Σειριακή_1

```

Στην ανάπτυξη αυτής της μεθόδου χρησιμοποιήθηκε μια μεταβλητή (done) που παίρνει την τιμή «Ψευδής» μόλις βρεθεί το αναζητούμενο στοιχείο παίρνει την τιμή «Αληθής» και με την εντολή position ← I καταγράφεται η θέση του στοιχείου. Και σταματά η διαδικασία. Αν δεν βρεθεί τότε η μεταβλητή done θα συνεχίσει να έχει την τιμή «Ψευδής» και η μεταβλητή position την τιμή 0

2. Σειριακή αναζήτηση σε μη ταξινομημένο πίνακα , κάποια στοιχεία του πίνακα εμφανίζονται παραπάνω από μία φορά

Αν κάποιο στοιχείο εμφανίζεται στον πίνακα παραπάνω από μία φορά , τότε ο αλγόριθμος πρέπει να τροποποιηθεί η μεταβλητή done είναι περιττή. Σ' αυτή τη περίπτωση η αναζήτηση συνεχίζεται μέχρι το τέλος του πίνακα αφού η μοναδική συνθήκη θα είναι $i \leq N$.

```

Αλγόριθμος Σειριακή_2
Δεδομένα // N, A, key //
position ← 0
i ← 1
Όσο i ≤ N επανάλαβε
    Αν A[i] = key τότε
        position ← i
    αλλιώς
        i ← i + 1
Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // done, position //
Τέλος Σειριακή_2

```

Προφανώς αν κάποιο στοιχείο του πίνακα επαναλαμβάνεται για παράδειγμα τρεις φορές θα μας δώσει τη θέση του τρίτου

3. Σειριακή αναζήτηση σε ταξινομημένο πίνακα

```

Αλγόριθμος Σειριακή_3
Δεδομένα // N, Table ,key //
done ← Ψευδής
position ← 0
i ← 1

```

```

Όσο (done=Ψευδής) και (i<=N) και (key>table[i]) επανάλαβε
  Αν table[i]=key τότε
    done ← Αληθής
    position ← i
    αλλιώς
      i ← i+1
  Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // done,position//
Τέλος Σειριακή_3

```

Στο παραπάνω αλγόριθμο η επανάληψη σταματά όταν 1) Το στοιχείο Βρεθεί 2) Όταν εξαντληθεί ο πίνακας και το στοιχείο δεν βρέθηκε και 3) Όταν το στοιχείο που αναζητάμε είναι μεγαλύτερο από κάποιο στοιχείο του πίνακα.

Ταξινόμηση στοιχείων πίνακα

Η διάταξη των στοιχείων ενός πίνακα κατά αύξουσα ή φθίνουσα σειρά (ταξινόμηση-Sorting) είναι μια πολύ σπουδαία εργασία και πάρα πολύ χρήσιμη σε ζητήματα όχι μόνο της καθημερινής ζωής αλλά και στα περισσότερα θέματα πληροφορικής. Μπορούμε να ταξινομήσουμε πίνακες που περιέχουν αριθμούς, αλλά και αλφαριθμητικά (π.χ Ονόματα και επίθετα)

Η Ταξινόμηση στοιχείων ενός πίνακα μας κάνει την αναζήτηση στοιχείων πίνακα ευκολότερη και γρηγορότερη.

Ταξινόμηση στοιχείων πίνακα είναι η διαδικασία της διάταξης των στοιχείων σε αύξουσα ή φθίνουσα σειρά.

Υπάρχουν πολλοί μέθοδοι ταξινόμησης και η κάθε μια έχει τα πλεονεκτήματα ή τα μειονεκτήματά της. Εμείς θα ασχοληθούμε παρακάτω με τη με την μέθοδο «**Ευθείας ανταλλαγής**» ή μέθοδος της «**Φυσαλίδας**» (Bubble sort) Η μεγάλη δημοτικότητα της προέρχεται από το παράξενο όνομά της, αλλά και από την απλότητά της. Έχει το χαρακτηριστικό της απλότητας δεν θεωρείται όμως αρκετά γρήγορη, ιδιαίτερα σε μεγάλους πίνακες.

Η μέθοδος αυτή στηρίζεται στην ιδέα ότι μπορούμε να συγκρίνουμε δύο γειτονικά στοιχεία του πίνακα και αν το πρώτο έχει χαμηλότερη αξία από το δεύτερο, τα εναλλάσσουμε, διαφορετικά τα αφήνουμε στη θέση τους.

Πιο συγκεκριμένα :

Στο πρώτο σάρωμα του πίνακα (π.χ $A[n]$) συγκρίνουμε το $A[10]$ με $A[9]$ στοιχείο και αν $A[10] > A[9]$ τότε τα αφήνουμε στη θέση τους, αν όμως $A[10] < A[9]$ τότε τους αλλάζουμε τη θέση τους. Αυτό συνεχίζεται για τα στοιχεία $A[9]$ και $A[8]$ κ.ο.κ. Η διαδικασία σταματά όταν δεν έχω να κάνω άλλες εναλλαγές.

Εφαρμόζουμε τα παραπάνω στον πίνακα $A=[18 \ 15 \ 23 \ 12 \ 5 \ 7 \ 13 \ 28 \ 14 \ 10]$

Αρχική Τιμή	ΣΑΡΩΜΑΤΑ ΠΙΝΑΚΑ								
	1	2	3	4	5	6	7	8	

A[1]	18	5	5	5	5	5	5	5	5
A[2]	15	18	7	7	7	7	7	7	7
A[3]	23	15	18	12	10	10	10	10	10
A[4]	12	23	15	18	12	12	12	12	12
A[5]	5	12	23	15	18	13	13	13	13
A[6]	7	7	12	23	15	18	14	14	14
A[7]	13	10	10	10	23	15	18	15	15
A[8]	28	13	13	13	13	23	15	18	18
A[9]	14	28	14	14	14	14	23	23	23
A[10]	10	14	28	28	28	28	28	28	28

ΠΑΡΑΤΗΡΗΣΕΙΣ

☼ Όπως φαίνεται από τον παραπάνω πίνακα (παρατηρήστε τους έντονα γραμμένους αριθμούς) οι τιμές με τη μεγαλύτερη αξία (βαρύτερες) τείνουν να κατεβαίνουν προς τα κάτω, ενώ οι τιμές με τη μικρότερη αξία τείνουν να ανεβαίνουν προς τα επάνω σαν φυσαλίδες, Έτσι έχει πάρει και το όνομά της.

☼ Η μέθοδος σταματά όταν δεν υπάρξει καμία εναλλαγή.

Ο παρακάτω αλγόριθμος μας υλοποιεί την παραπάνω μέθοδο.

Αλγόριθμος Bubble_Sort

plithos ← 10

! Διάβασμα του πίνακα

Για χ από 1 μέχρι plithos

 Διάβασε A[x]

Τέλος επανάληψης

! Αρχή μεθόδου

Για I από 2 μέχρι plithos

 για j από plithos μέχρι I με βήμα -1

 Αν A[j-1] > A[j] τότε

 αντιμετάθεσε A[j-1], A[j]

τέλος αν

τέλος επανάληψης

Τέλος επανάληψης

! Αποτελέσματα

Για χ από 1 μέχρι plithos

 τύπωσε A[x]

Τέλος επανάληψης

Τέλος Bubble_Sort

☼ Στον παραπάνω πίνακα ταξινομήσαμε ακέραιους. Ο ακέραιος τύπος επιλέχτηκε χάριν απλότητας. Θα μπορούσαμε να ταξινομήσουμε πραγματικούς τύπο αριθμών ή και τύπο χαρακτήρα (Γράμματα- Λέξεις -string)

☼ Η εντολή «αντιμετάθεσε» στον παραπάνω αλγόριθμο θα μπορούσε να υλοποιηθεί χρησιμοποιώντας μια βοηθητική μεταβλητή (temp)

temp ← A[j-1]

A[j-1] ← A[j]

A[j] ← temp

Παρατήρηση

Σαν απάντηση στην άσκηση του Τετραδίου του μαθητή (ΔΤ2 σελίδα 33)

Ο Αλγόριθμος της φουσαλίδας δεν είναι αρκετά «έξυπνος» ώστε να διαπιστώνει στην αρχή ή στο τέλος αν ο πίνακας είναι ταξινομημένος, ώστε να αποφεύγονται τα περιττά περάσματα (που σε συνολικό αριθμό είναι $N-1$).

Για τη λύση του παραπάνω προβλήματος θα χρησιμοποιηθεί μια λογική μεταβλητή (Flag), που πριν από κάθε πέρασμα **αρχικοποιείται ως ψευδής** και αλλάζει ως αληθής αν σε κάποιο πέρασμα γίνει έστω και μία ανταλλαγή. Έτσι αν σε κάποιο πέρασμα δεν γίνει καμία ανταλλαγή, τότε η flag παραμένει ψευδής και αυτομάτως τελειώνει ο αλγόριθμος,

Αλγόριθμος φουσαλίδα_2

Δεδομένα // table//

Αρχή επανάληψης

Flag ← ψευδής

Για i από 1 μέχρι $n-1$

Αν $table[i+1] < table[i]$ τότε

αντιμετάθεσε $table[i+1]$, $table[i]$

flag ← αληθής

Τέλος_αν

Τέλος_επανάληψης

Μέχρις_ότου flag=ψευδής

Αποτελέσματα // table//

τέλος Φουσαλίδας_2

Συγχώνευση δύο πινάκων

Η συγχώνευση είναι μια ακόμα βασική εργασία σε πίνακες. Ο σκοπός της συγχώνευσης δύο ή περισσότερων πινάκων, είναι η δημιουργία ενός άλλου πίνακα με διπλάσιο αριθμό στηλών.

Πρόβλημα 1: Δίνονται δύο πίνακες γραμμή A και B

$A = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6]$ και $B = [b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6]$

Ζητείται να δημιουργηθεί τρίτος πίνακας

$\Gamma = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6]$

Λύση του προβλήματος 1

Αλγόριθμος Συγχώνευση_1

Δεδομένα // A, B //

Για i από 1 μέχρι 6

$\Gamma[i] \leftarrow A[i]$

$\Gamma[i+6] \leftarrow B[i]$

Τέλος_επανάληψης

Τέλος_επανάληψης

Αποτελέσματα // Γ //

Τέλος Συγχώνευση_1

Πρόβλημα 2 : Να γραφεί αλγόριθμος που να συγχωνεύει (συνενώνει) δύο δεδομένους πίνακες A και B διαστάσεων 5x5, σε έναν τρίτο πίνακα διαστάσεων 5x10.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & \vdots & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & \vdots & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & \vdots & b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & \vdots & b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & \vdots & b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix}$$

Παρατήρηση: Για τη συγκεκριμένη άσκηση υπάρχουν πολλές παραλλαγές, π.χ. ο πίνακας B να έχει διαστάσεις 5x7, ή να δίνεται ο πίνακας Γ και να ζητείται ο διαχωρισμός του σε δύο άλλους πίνακες κατάλληλων διαστάσεων, κ.ο.κ.

Αλγόριθμος Συγχώνευση_2

Δεδομένα // A, B //

Για i από 1 μέχρι 5

Για j από 1 μέχρι 5

 Γ[i, j] ← A[i, j]

 Γ[i, j + 5] ← B[i, j]

Τέλος_επανάληψης

Τέλος_επανάληψης

Αποτελέσματα // Γ //

Τέλος Συγχώνευση_2

Τώρα τίθεται το ερώτημα : Μπορούμε να συνενώσουμε δύο ταξινομημένους πίνακες. Και δημιουργήσουμε ένα άλλο πίνακα που να είναι και αυτός ταξινομημένος ;

Παρακάτω θα δοθεί ο αλγόριθμος συγχώνευσης δύο ταξινομημένων πινάκων και η δημιουργία ενός τρίτου πίνακα επίσης ταξινομημένου. Θα θεωρήσουμε σαν δεδομένα τους πίνακες A[] και B[] και το πλήθος των στοιχείων τους m και n αντίστοιχα.

Η μέθοδος στηρίζεται στην ιδέα ότι : Συγκρίνουμε A[1]<B[1] αν ναι τοποθετούμε στον νέο πίνακα Γ[1] ← A[1] αν όχι Γ[1] ← B[1] και η διαδικασία συνεχίζεται μέχρι να τελειώσει ο ένας πίνακας από τους δεδομένους. Η συνέχιση της διαδικασίας απαιτεί να χρησιμοποιήσουμε τρεις δείκτες I, J, K για να παρακολουθούμε σε ποιο σημείο βρισκόμαστε μέσα σε κάθε πίνακα

.Αφού τελειώσει ο ένας πίνακας αντιγράφουμε τα υπόλοιπα στοιχεία του άλλου στον τρίτο πίνακα Γ[]

Αλγόριθμος Συνένωση
Δεδομένα // A[], B[], n, m//
DA ← 1
DB ← 1
DG ← 1
Όσο (DA ≤ n) ΚΑΙ (DB ≤ m) **επανάλαβε**
 Αν A[DA] < B[DB] **τότε**
 Γ[DG] ← A[DA]
 DA ← DA + 1
 αλλιώς
 Γ[DG] ← B[DB]
 DB ← DB + 1
 Τέλος_αν
 DG ← DG + 1
Τέλος_επανάληψης
Αν DA > n **τότε**
 Για t από DB **μέχρι** m
 Γ[DG + t - DB] ← B[t]
 τέλος_επανάληψης
 αλλιώς
 Για t από DA **μέχρι** n
 Γ[DG + t - DA] ← A[t]
 Τέλος_επανάληψης
Τέλος_αν
Αποτελέσματα // Γ[]//
Τέλος Συγχώνευση

4. 2^η Μορφή δομής δεδομένων ΕΓΓΡΑΦΗ (Record)

Ένα σημαντικό μειονέκτημα των πινάκων είναι ότι υποχρεωτικά όλα του τα στοιχεία πρέπει να είναι του ίδιου τύπου (Ακέραιος, πραγματικός, τύπος χαρακτήρα κ.λ.π). Το πρόβλημα αυτό λύνεται με την 2^η μορφή δεδομένων «**Εγγραφή-Record**» Θα παρουσιάσουμε τη νέα δομή δεδομένων με ένα παράδειγμα.

Πολλές φορές στο σχολείο σας χρειάστηκε να φτιάξετε καταστάσεις που να περιέχουν το σύνολο των μαθητών του τμήματος και σε διπλανές στήλες να βάζετε ημερομηνίες ή κάποιο ποσό χρημάτων, όπως

ΑΡ.ΜΗΤΡΩΟΥ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΗΜΕΡΟΜΗΝΙΑ	ΠΟΣΟ €
234	ΣΗΦΑΚΑΚΗΣ	ΝΙΚΟΣ	12-11-04	120
286	ΚΡΗΤΙΚΑΚΗΣ	ΠΟΛΟΣ	10-11-04	110
.....				

.....				
286	ΝΙΚΟΛΟΥΔΑΚΗΣ	ΣΙΦΗΣ	10-10-04	89

Παρατηρούμε ότι κάθε γραμμή είναι ένα σύνολο πληροφοριών που αναφέρονται στην οντότητα μαθητής. Θα λέμε ότι κάθε γραμμή που αποτελεί μια **εγγραφή (Record)**. Η κάθε στήλη μπορεί να είναι διαφορετικού τύπου και ονομάζεται **πεδίο (field)** της εγγραφής. Έτσι η πρώτη γραμμή (πρώτη εγγραφή) περιγράφει την οντότητα «ΣΗΦΑΚΑΚΗΣ ΝΙΚΟΣ» και αποτελείται από τα πεδία 1) ΑΡ.ΜΗΤΡΩΟΥ 2) ΕΠΙΘΕΤΟ 3) ΟΝΟΜΑ 4) ΗΜΕΡΟΜΗΝΙΑ 5) ΠΟΣΟ. Η διάταξη αυτή της εγγραφής ονομάζεται **γραμμογράφηση** της εγγραφής

Γενικά λοιπόν μπορούμε να πούμε ότι :

Εγγραφή είναι ένα σύνολο αλληλοσυσχετιζόμενων πεδίων που περιγράφουν μια οντότητα.

Ένα σύνολο εγγραφών λέμε ότι δημιουργεί μια νέα δομή δεδομένων που ονομάζεται **Αρχείο (Δυναμική δομή δεδομένων)**

☼ Σε ένα αρχείο δεν είναι προκαθορισμένο το μέγεθός του κατά τη στιγμή του προγραμματισμού, τα στοιχεία αυτής της δομής δεν αποθηκεύονται σε συνεχόμενες θέσεις της βοηθητικής μνήμης.

5. 3^η Μορφή δομής Δεδομένων ΣΤΟΙΒΑ (Stack)

(Από Βιβλίο Σελίδα 59 μέχρι 60)

6. 4^η Μορφή δομής δεδομένων ΟΥΡΑ (Queue)

(Από βιβλίο σελίδα 60 μέχρι 63)



ΑΣΚΗΣΕΙΣ ΣΤΟΥΣ ΠΙΝΑΚΕΣ

ΑΣΚΗΣΗ 1 Να γραφεί αλγόριθμος ο οποίος να διαβάσει ένα μονοδιάστατο πίνακα ακέραιων αριθμών A με N θέσεις και υπολογίζει και να τυπώνει το μέσο όρο, το άθροισμα και το γινόμενο των στοιχείων του πίνακα A .

ΑΣΚΗΣΗ 2 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα μονοδιάστατο πίνακα ακέραιων αριθμών A με 10 θέσεις και υπολογίζει και να τυπώνει το μέγιστο και το ελάχιστο στοιχείο του πίνακα A και τις θέσεις τους.

ΑΣΚΗΣΗ 3 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα μονοδιάστατο πίνακα ακέραιων αριθμών A με N θέσεις και υπολογίζει και να τυπώνει το μέσο όρο, και πόσοι από τους αριθμούς είναι μεγαλύτεροι από τον πιο πάνω μέσο όρο.

ΑΣΚΗΣΗ 4 Να γραφεί αλγόριθμος ο οποίος να δημιουργεί και να εμφανίζει μονοδιάστατο πίνακα A 100 θέσεων σύμφωνα με την σχέση:

$$A_i = \begin{cases} (2 * x)^i, & \text{όταν } x < 10 \\ i^x, & \text{διαφορετικά} \end{cases}$$

ΑΣΚΗΣΗ 5 Να γραφεί αλγόριθμος ο οποίος να διαβάζει δύο μονοδιάστατους πίνακες με όνομα A και B 5 θέσεων και να εμφανίζει το πίνακα C ο οποίος είναι το άθροισμα των A και B .

ΑΣΚΗΣΗ 6 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα πίνακα ακέραιων αριθμών A , δύο διαστάσεων $N \times M$ και να υπολογίζει και να τυπώνει το μέσο όρο, το άθροισμα και το γινόμενο των στοιχείων του πίνακα A .

ΑΣΚΗΣΗ 7 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα πίνακα ακέραιων αριθμών A , δύο διαστάσεων $N \times M$ και να υπολογίζει και να τυπώνει το μέγιστο και ελάχιστο στοιχείο του πίνακα καθώς και τις θέσεις τους στο πίνακα.

ΑΣΚΗΣΗ 8 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα πίνακα ακέραιων αριθμών A , δύο διαστάσεων 3×3 και να υπολογίζει και να τυπώνει α) το μέγιστο στοιχείο (και τη θέση του) της 3ης γραμμής β) το ελάχιστο στοιχείο (και τη θέση του) της 2ης στήλης γ) το άθροισμα των στοιχείων της 1ης γραμμής δ) το γινόμενο των στοιχείων της 3ης στήλης και ε) το άθροισμα των στοιχείων της κυρίας διαγωνίου.

$$\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}$$

Τα στοιχεία της κυρίας διαγωνίου είναι τα: a_{11} , a_{22} και a_{33} .

ΑΣΚΗΣΗ 9 Να γραφεί αλγόριθμος ο οποίος να διαβάζει ένα πίνακα ακέραιων αριθμών A , δύο διαστάσεων 3×3 και να υπολογίζει και να τυπώνει το άθροισμα των στοιχείων της δευτερεύουσας διαγωνίου.

$$\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}$$

Τα στοιχεία της δευτερεύουσας διαγωνίου είναι τα: a_{13} , a_{22} και a_{31} .

ΑΣΚΗΣΗ 10 Να γραφεί αλγόριθμος ο οποίος να κατασκευάζει δύο μονοδιάστατους πίνακες με όνομα ODD και EVEN αντίστοιχα ως εξής. Ο πρώτος να περιέχει τους περιττούς αριθμούς από το 1 ως το 100 και ο δεύτερος τους άρτιους αριθμούς

ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ-αλγόριθμοι

ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ-ΔΟΜΗ ΑΚΟΛΟΥΘΙΑΣ

1. Αλγόριθμος Παράσταση_E1

! Εισαγωγή δεδομένων

A ← 2

B ← 1

C ← 4

! Υπολογισμοί

E1 ← $(B+C)^A / (C-A)$

! Αποτελέσματα

Τύπωσε E1

Τέλος Παράσταση_E1

2. Αλγόριθμος Υπολογισμοί

! Εισαγωγή δεδομένων

Διάβασε A

Διάβασε B

Διάβασε C

Διάβασε D

! Υπολογισμοί
 SUM ← A+B+C+D
 PRODUCT ← A*B*C*D
 AVER ← SUM/4
 ! Αποτελέσματα
 Εκτύπωσε SUM
 Εκτύπωσε PRODUCT
 Εκτύπωσε AVER
 Τέλος Υπολογισμοί

ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ -ΔΟΜΗ ΕΠΙΛΟΓΗΣ

1) Αλγόριθμος Αριθμός_Θετικός1

Διάβασε Ar

Αν Ar > 0 τότε

 sxolio ← “ Ο Αριθμός είναι θετικός»

Τέλος_αν

Τύπωσε sxolio

Τέλος Αριθμός_Θετικός1

2) Αλγόριθμος Αριθμός_Θετικός2

Διάβασε Ar

Αν Ar > 0 τότε

 sxolio ← “ Ο Αριθμός είναι θετικός»

 αλλιώς

 sxolio ← “ Ο Αριθμός είναι αρνητικός»

Τέλος_αν

Τύπωσε sxolio

Τέλος Αριθμός_Θετικός2

3) Αλγόριθμος Βαθμός_Μαθητή

! Εισαγωγή δεδομένων

Γράψε «Δώσε ένα γράμμα A , B, C , D»

Διάβασε Xaract

! Επεξεργασία δεδομένων

Επίλεξε Xaract

 Περίπτωση “A”

 Bathmos ← “ΑΡΙΣΤΑ»

 Περίπτωση “B”

 Bathmos ← “ΚΑΛΑ»

 Περίπτωση “C”

 Bathmos ← “ΜΕΤΡΙΑ»

 Περίπτωση “D”

 Bathmos ← “ΑΠΕΤΥΧΕ»

 Περίπτωση αλλιώς

 Bathmos ← “ΕΔΩΣΕΣ ΛΑΘΟΣ ΧΑΡΑΚΤΗΡΑ»

Τέλος_επιλογών

! Αποτελέσματα

Τύπωσε Bathmos

Τέλος Βαθμός_Μαθητή

ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ ΣΤΗΝ ΕΜΦΩΛΕΥΜΕΝΗ ΕΠΙΛΟΓΗ

ΑΣΚΗΣΗ 1

Αλγόριθμος Μηνιαίος_Λογαριασμός

Διάβασε ΧΑ

Διάβασε ΣΔ

Διάβασε ΚΝ

ΠΑΓΙΟ ← 0

Αν ΚΝ > 0 τότε

 Αν ΚΝ ≤ 20 τότε

 ΚΟΣΤΟΣ ← (ΚΝ * 60)

 Αλλιώς

 ΚΟΣΤΟΣ ← (20 * 60) + (ΚΝ - 20) * 150

 Τέλος_αν

Τέλος_Αν

ΣΥΝ-ΚΟΣΤΟΣ ← ΠΑΓΙΟ + ΚΟΣΤΟΣ

Τύπωσε ΣΥΝ-ΚΟΣΤΟΣ

Τέλος Μηνιαίος_Λογαριασμός

ΑΣΚΗΣΗ 2

Μια λύση του παραπάνω προβλήματος είναι η παρακάτω :

Αλγόριθμος MAX.1

Διάβασε Α

Διάβασε Β

Διάβασε C

Αν (Α > Β) ΚΑΙ (Α > C) τότε

 Max ← Α

Τέλος_Αν

Αν (Β > Α) ΚΑΙ (Β > C) τότε

 Max ← Β

Τέλος_αν

Αν C > Α ΚΑΙ C > Β Τότε

 Max ← C

Τέλος_Αν

Τύπωσε MAX

Τέλος MAX1

Η λύση αυτή δεν είναι η καλύτερη, διότι χρησιμοποιούμε τρία διαφορετικά Αν...τότε. Δείτε την παρακάτω λύση

Αλγόριθμος MAX2.

Διάβασε Α

Διάβασε Β

Διάβασε C

Αν B > A τότε

 Αν C > B τότε

 MAX ← C

 αλλιώς

 MAX ← B


```

    Τέλος_Αν
αλλιώς
    Αν C>A τότε
        MAX←C
    Αλλιώς
        MAX←A
    Τέλος_Αν
τέλος_αν
Τύπωσε MAX
Τέλος MAX2

```

Μια Τρίτη εναλλακτική λύση είναι η παρακάτω

```

Αλγόριθμος Max3
Δεδομένα //A,B,C//
Αν A>B και A>C τότε
    MAX←A
αλλιώς_αν B>A και B>C τότε
    MAX ← B
Αλλιώς
    MAX ←C
τέλος_αν
Αποτελέσματα // MAX//
Τέλος Max3

```

Στα παραπάνω παραδείγματα φαίνεται η διαφορά στη χρήση των διαφόρων μορφών εντολών Αν... τότε . Προφανώς μπορούν να υλοποιηθούν και άλλοι αλγόριθμοι για την επίλυση του ίδιου προβλήματος.

Λύση άσκησης 4

Αλγόριθμος Υπολογισμοί

Διάβασε A ,B , C

Max ← A

Αν B>C τότε

AFAIRESH ← B-C

Αλλιώς

AFAIRESH← C-B

Τέλος_Αν

Αν B>MAX τότε

Max ← B

Αν A>C τότε

AFAIRESH ← A-C

Αλλιώς

AFAIRESH ← C-A

Τέλος_Αν

αλλιώς

Αν C> MAX τότε

MAX← C

Αν A>B τότε

AFAIRESH← A-B

Αλλιώς

AFAIRESH ← B-A

Τέλος_Αν

Τέλος_Αν

Τέλος_Αν

Αποτ ← MAX*AFAIRESH

Τύπωσε Αποτ

Τέλος Υπολογισμοί

ΛΥΣΗ ΑΣΚΗΣΗΣ 5

Αλγόριθμος ασκηση_5

Δεδομένα // B1,B2,β3,β4//

Αν (B1>=8) ΚΑΙ (B2>=8) ΚΑΙ (B3.=8) ΚΑΙ (B4>=8) τότε

ΜΟ ← (B1+B2+B3+B4)/4

Αν ΜΟ>=10 τότε

Τύπωσε «ΠΡΟΑΓΕΤΑΙ»

Αλλιώς

Τύπωσε «ΑΠΟΡΡΙΠΤΕΤΑΙ»

Τέλος_αν

Αλλιώς

Τύπωσε «ΚΑΠΟΙΟΣ ΒΑΘΜΟΣ <8»

Τέλος_Αν

Τέλος ασκηση_5



ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ ΣΤΙΣ ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΔΟΜΕΣ

ΛΥΣΗ ΑΣΚΗΣΗΣ 1

Αλγόριθμος Αθροισμα1

! Εισαγωγή δεδομένων

Διάβασε N

! Επεξεργασία δεδομένων

sum ← 0

gin ← 1

Για χ από 1 μέχρι N

sum ← sum+x

gin ← gin*x

τέλος επανάληψης

mo ← sum/n

! Αποτελέσματα

Τύπωσε mo

Τύπωσε sum

Τύπωσε gin

Τέλος Αθροισμα1

ΛΥΣΗ ΑΣΚΗΣΗΣ 2

Αλγόριθμος Αθροισμα2
 ! Εισαγωγή δεδομένων
Διάβασε N
 ! Επεξεργασία δεδομένων
 $m \leftarrow 0$
 $sum \leftarrow 0$
 $gin \leftarrow 1$
Για x **από** 1 **μέχρι** N **βήμα** 2
 $m \leftarrow m+1$
 $sum \leftarrow sum+x$
 $gin \leftarrow gin*x$
τέλος επανάληψης
 $mo \leftarrow sum/m$
 ! Αποτελέσματα
Τύπωσε mo
Τύπωσε sum
Τύπωσε gin
Τέλος Αθροισμα2

ΛΥΣΗ ΑΣΚΗΣΗΣ 3

Αλγόριθμος Αθροισμα3
 ! Εισαγωγή δεδομένων
Διάβασε N
 ! Επεξεργασία δεδομένων
 $m \leftarrow 0$
 $sum \leftarrow 0$
 $gin \leftarrow 1$
Για x **από** 2 **μέχρι** N **βήμα** 2
 $sum \leftarrow sum+x$
 $gin \leftarrow gin*x$
 $m \leftarrow m+1$
τέλος επανάληψης
 $mo \leftarrow sum/m$
 ! Αποτελέσματα
Τύπωσε mo
Τύπωσε sum
Τύπωσε gin
Τέλος Αθροισμα3



ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ στις ΜΕΤΑΤΡΟΠΕΣ αλγορίθμων

Λύση άσκησης 1

Αλγόριθμος Εμφωλευμένες
Διάβασε α
Αν $\alpha < 0$ **τότε**
 εμφάνισε “Αρνητικό”

αλλιώς_αν $\alpha \leq 100$ τότε
 εμφάνισε “Μεταξύ 0 και 100”
αλλιώς_αν $\alpha \leq 200$ τότε
 εμφάνισε “Μεταξύ 100 και 200”
αλλιώς
 εμφάνισε “Μεγαλύτερο από 200”
Τέλος_αν
Τέλος Εμφωλευμένες