

**ΘΕΜΑ Α**

**A1.** Να γράψετε στο τετράδιό σας τον αριθμό καθεμιάς από τις παρακάτω προτάσεις **1** έως **5** και δίπλα τη λέξη **ΣΩΣΤΟ**, αν η πρόταση είναι σωστή, ή τη λέξη **ΛΑΘΟΣ**, αν η πρόταση είναι λανθασμένη.

1. Ο δείκτης σε έναν πίνακα έχει υποχρεωτικά ακέραια τιμή.
2. Η έκφραση  $X \text{ ΚΑΙ } (\text{ΟΧΙ } X)$  είναι πάντα Αληθής για κάθε τιμή της λογικής μεταβλητής  $X$ .
3. Η έκφραση "**ΚΑΛΗΜΕΡΑ**" > "**ΚΑΛΗΣΠΕΡΑ**" έχει την τιμή Αληθής.
4. Σκοπός της ιεραρχικής σχεδίασης είναι η διάσπαση του προβλήματος σε μια σειρά από απλούστερα υποπροβλήματα.
5. Στη ΓΛΩΣΣΑ έχουμε απεριόριστη εμβέλεια μεταβλητών.

**A2.** Να αναφέρετε και να περιγράψετε τέσσερις από τις βασικές λειτουργίες επί των δομών δεδομένων που μπορούν να χρησιμοποιηθούν στους πίνακες.

**A3.** Δίνεται το παρακάτω τμήμα προγράμματος:

$X \leftarrow K$

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

$X \leftarrow X + 2$

**ΓΡΑΨΕ X**

**ΜΕΧΡΙΣ\_ΟΤΟΥ**  $X > = M$

Τι θα εμφανίσει για κάθε μία από τις παρακάτω περιπτώσεις:

α)  $K = 4, M = 9$

β)  $K = 5, M = 0$

γ)  $K = -1, M = 3$

**A4.** Να σχεδιάσετε τα διαγράμματα ροής για τις παρακάτω εντολές επανάληψης:

**α)** ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ

εντολές

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

**β)** ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

εντολές

ΜΕΧΡΙΣ\_ΟΤΟΥ συνθήκη

**A5.** Δίνεται ο παρακάτω αλγόριθμος πολλαπλασιασμού αλλά ρωσικά δύο θετικών ακεραίων αριθμών  $M_1$  και  $M_2$  σε φυσική γλώσσα κατά βήματα:

Βήμα 1      Θέσε  $P = 0$

Βήμα 2      Αν  $M_2 > 0$ , τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7 Αν ο  $M_2$  είναι

Βήμα 3      περιττός, τότε θέσε  $P = P + M_1$

Βήμα 4      Θέσε  $M_1 = M_1 * 2$

Βήμα 5      Θέσε  $M_2 = M_2 / 2$  (θεώρησε μόνο το ακέραιο μέρος) Πήγαινε στο Βήμα 2

Βήμα 6      Τύπωσε τον  $P$

Βήμα 7

Να γράψετε στο τετράδιό σας την κωδικοποίηση των παραπάνω βημάτων σε ΓΛΩΣΣΑ.

## ΘΕΜΑ Β

**B1.** Δίνεται ο παρακάτω αλγόριθμος ο οποίος ελέγχει αν το στοιχείο key βρίσκεται στον πίνακα table[n] τουλάχιστον τρεις (3) φορές και εμφανίζει τη θέση στην οποία βρίσκεται την τρίτη φορά.

**Αλγόριθμος B1**

**Δεδομένα** // n, table, key //

done  $\leftarrow$  ψευδής

position  $\leftarrow$  0

i  $\leftarrow$  1

count  $\leftarrow$  ...**(1)**...

**Όσο** i  $\leq$  ...**(2)**... **και** done = ...**(3)**... **επανάλαβε**

**Αν** table[ ...**(4)**... ] = key **τότε**

        count  $\leftarrow$  ...**(5)**...

**Τέλος\_αν**

**Αν** count = ...**(6)**... **τότε**

        done  $\leftarrow$  ...**(7)**...

        ...**(8)**...  $\leftarrow$  i

**αλλιώς**

        i  $\leftarrow$  ...**(9)**...

**Τέλος\_αν**

**Τέλος\_επανάληψης**

**Αν** ...**(10)**... **τότε**

**Εμφάνισε** "Το στοιχείο", key, "υπάρχει τουλάχιστον 3 φορές."

**Εμφάνισε** "Για τρίτη φορά εμφανίζεται στη θέση ", position, "."

**αλλιώς**

**Εμφάνισε** "Το στοιχείο", key, "δεν υπάρχει τουλάχιστον 3 φορές."

**Τέλος\_αν Τέλος B1**

Να γράψετε στο τετράδιό σας τους αριθμούς των κενών και δίπλα ό,τι χρειάζεται να συμπληρωθεί έτσι ώστε ο αλγόριθμος να λειτουργεί σωστά.

**B2.** Δίνονται οι παρακάτω δηλώσεις υποπρογραμμάτων και των παραμέτρων τους:

ΣΥΝΑΡΤΗΣΗ A( $\chi$ ,  $\psi$ ): ΠΡΑΓΜΑΤΙΚΗ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:  $\chi$

ΧΑΡΑΚΤΗΡΕΣ:  $\psi$ [10]

ΔΙΑΔΙΚΑΣΙΑ B( $\chi$ ,  $\psi$ ,  $\zeta$ )

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:  $\psi$

ΠΡΑΓΜΑΤΙΚΕΣ:  $\chi$ ,  $\zeta$

Επίσης δίνεται το τμήμα δηλώσεων κύριου προγράμματος:

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ:  $\kappa$ ,  $\lambda$ [10],  $\mu$

ΠΡΑΓΜΑΤΙΚΕΣ:  $\pi$ ,  $\rho$ [10],  $\gamma$  ΧΑΡΑΚΤΗΡΕΣ:  $\theta$ [10],  $\nu$

Μεταξύ των εντολών του κύριου προγράμματος υπάρχουν οι παρακάτω πέντε εντολές κλήσης των υποπρογραμμάτων:

1.  $\pi \leftarrow A(\kappa, \nu)$
2. ΚΑΛΕΣΕ A( $\mu, \theta$ )
3. ΚΑΛΕΣΕ B( $\pi, \mu$ )
4.  $\nu \leftarrow A(\mu, \theta)$
5.  $\kappa \leftarrow B(\pi, \mu, \rho[1])$

Καθεμιά από τις παραπάνω εντολές έχει ένα λάθος.

- α) Να γράψετε στο τετράδιό σας τον αριθμό (1 – 5) της καθεμιάς εντολής και δίπλα να περιγράψετε το λάθος.
- β) Να γράψετε στο τετράδιό σας τον αριθμό (1 – 5) της καθεμιάς εντολής και δίπλα να γράψετε την εντολή σωστά χρησιμοποιώντας μόνο μεταβλητές που υπάρχουν στο τμήμα δηλώσεων του κύριου προγράμματος.

### ΘΕΜΑ Γ

Το Υπουργείο Παιδείας παρέχει μέσω του διαδικτύου μια συλλογή από εκπαιδευτικά βίντεο. Ο αριθμός των επισκέψεων που δέχεται κάθε ένα βίντεο καταγράφεται από ειδικό λογισμικό. Τα βίντεο διακρίνονται σε τρεις κατηγορίες ανάλογα με την επισκεψιμότητά τους, σύμφωνα με τον παρακάτω πίνακα:

ΚΑΤΗΓΟΡΙΕΣ ΕΠΙΣΚΕΨΙΜΟΤΗΤΑΣ	
Όνομα	Αριθμός Επισκέψεων
Χαμηλή	από 1 έως και 100
Μεσαία	από 101 έως και 1000
Υψηλή	πάνω από 1000

Τα βίντεο με μηδενικές επισκέψεις δεν κατατάσσονται σε καμία κατηγορία.

Να αναπτύξετε πρόγραμμα το οποίο:

- Γ1. Να περιλαμβάνει κατάλληλο τμήμα δηλώσεων.
- Γ2. Να διαβάζει επαναληπτικά τον τίτλο κάθε βίντεο και τον αριθμό των επισκέψεων που δέχτηκε. Η είσοδος των δεδομένων να τερματίζεται, όταν ως τίτλος βίντεο δοθεί η λέξη «ΤΕΛΟΣ».  
Να γίνεται έλεγχος εγκυρότητας ώστε ο αριθμός των επισκέψεων να μην είναι αρνητικός.
- Γ3. Να βρίσκει και να εμφανίζει τον τίτλο του βίντεο με τον μεγαλύτερο αριθμό επισκέψεων.  
Να θεωρήσετε ότι είναι μοναδικό.
- Γ4. Να υπολογίζει για καθεμιά από τις τρεις κατηγορίες επισκεψιμότητας το πλήθος των βίντεο που καταχωρίστηκαν σε αυτή. Να εμφανίζει για κάθε κατηγορία:
  - το όνομά της και
  - το πλήθος των βίντεο που περιλαμβάνει.
- Γ5. Να βρίσκει και να εμφανίζει το όνομα της κατηγορίας επισκεψιμότητας στην οποία καταχωρίστηκαν τα περισσότερα βίντεο. Να θεωρήσετε ότι είναι μοναδική.

Σημείωση

Το πλήθος των βίντεο δεν είναι γνωστό.

### **ΘΕΜΑ Δ**

Στην 27η Βαλκανιάδα Πληροφορικής που θα διεξαχθεί στην Αθήνα τον Σεπτέμβριο του 2019, συμμετέχουν 40 μαθητές. Κάθε μαθητής παίρνει έναν κωδικό από 1 έως και 40, ο οποίος αντιστοιχεί στη σειρά που δήλωσε συμμετοχή. Κάθε μαθητής καλείται να επιλύσει έξι προβλήματα. Για κάθε πρόβλημα αναπτύσσει τη λύση του σε μία γλώσσα προγραμματισμού και την υποβάλλει για βαθμολόγηση. Η λύση βαθμολογείται σε ακέραια κλίμακα από 0 έως 100.

Κατά τη διάρκεια του διαγωνισμού κάθε μαθητής και για κάθε πρόβλημα μπορεί να υποβάλλει τη λύση του όσες φορές θέλει.

Να αναπτύξετε πρόγραμμα σε ΓΛΩΣΣΑ το οποίο:

- Δ1.** Να περιλαμβάνει κατάλληλο τμήμα δηλώσεων.
- Δ2.** Να διαβάξει επαναληπτικά τα ονόματα των μαθητών και να τα καταχωρίζει στον Πίνακα **ON**[40]. Επίσης, να αρχικοποιεί με την τιμή 0 όλα τα στοιχεία του Πίνακα **BAΘ**[40,6], ο οποίος θα περιέχει τη βαθμολογία κάθε μαθητή για κάθε πρόβλημα.
- Δ3.** Κάθε φορά που μία λύση προβλήματος υποβάλλεται και βαθμολογείται, το πρόγραμμα να διαβάξει τον κωδικό του μαθητή (από 1 έως και 40), τον αριθμό του προβλήματος (από 1 έως και 6) και τη βαθμολογία του (από 0 έως και 100).  
Η βαθμολογία να καταχωρίζεται στην αντίστοιχη θέση του Πίνακα **BAΘ**[40,6] μόνο αν είναι μεγαλύτερη από τη βαθμολογία που είναι ήδη καταχωρισμένη.  
Για τον τερματισμό της εισαγωγής δεδομένων το πρόγραμμα να εμφανίζει το μήνυμα «Υπάρχει νέα λύση προβλήματος; **NAI / OXI**». Αν εισαχθεί η τιμή «**OXI**», να τερματίζεται η εισαγωγή δεδομένων.
- Δ4.** Να υπολογίζει και να καταχωρίζει στον Πίνακα **ΣB**[40] τα αθροίσματα των βαθμολογιών κάθε μαθητή στα έξι προβλήματα. Για τον σκοπό αυτό να καλεί μόνο μια φορά το υποπρόγραμμα με όνομα **ΥΣB**.  
Να αναπτύξετε το υποπρόγραμμα **ΥΣB** το οποίο να δέχεται ως είσοδο τον Πίνακα **BAΘ**[40,6] και να επιστρέφει ως έξοδο συμπληρωμένο τον Πίνακα **ΣB**[40].
- Δ5.** Να εμφανίζει τα ονόματα των μαθητών ταξινομημένων σύμφωνα με τη συνολική τους βαθμολογία σε φθίνουσα βαθμολογική σειρά. Σε περίπτωση μαθητών με την ίδια βαθμολογία, τα ονόματά τους να εμφανίζονται με αλφαβητική σειρά.

### Σημειώσεις

- α) Δεν απαιτούνται έλεγχοι εγκυρότητας τιμών.
- β) Να θεωρήσετε ότι θα δοθεί τουλάχιστον μια λύση προβλήματος από έναν μαθητή.