

**ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ: ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ**

**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  $X \text{ ΚΑΙ } (OXI \ X)$  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.

**Μονάδες 10**

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

**Μονάδες 8**

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

$X \leftarrow K$

**ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ**

$X \leftarrow X + 2$

**ΓΡΑΨΕ X**

**ΜΕΧΡΙΣ\_ΟΤΟΥ**  $X \geq M$

Τι θα εμφανίσει για κάθε μία από τις παρακάτω περιπτώσεις:

α)  $K = 4, M = 9$

β)  $K = 5, M = 0$

γ)  $K = -1, M = 3$

**Μονάδες 6**

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

**α)** ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ

εντολές

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

(μονάδες 3)

**β)** ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

εντολές

ΜΕΧΡΙΣ\_ΟΤΟΥ συνθήκη

(μονάδες 3)

**Μονάδες 6**

**A5.** Δίνεται ο παρακάτω αλγόριθμος πολλαπλασιασμού αλά ρωσικά δύο θετικών ακεραίων αριθμών  $M1$  και  $M2$  σε φυσική γλώσσα κατά βήματα:

Βήμα 1 Θέσε  $P=0$

Βήμα 2 Αν  $M2>0$ , τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7

Βήμα 3 Αν ο  $M2$  είναι περιττός, τότε θέσε  $P=P+M1$

Βήμα 4 Θέσε  $M1=M1*2$

Βήμα 5 Θέσε  $M2=M2/2$  (θεώρησε μόνο το ακέραιο μέρος)

Βήμα 6 Πήγαινε στο Βήμα 2

Βήμα 7 Τύπωσε τον  $P$

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

**Μονάδες 10**

## 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 ← ψευδής

position ← 0

i ← 1

count ← ...**(1)**...

**Όσο** i <= ...**(2)**... **και** done = ...**(3)**... **επανάλαβε**

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

count ← ...**(5)**...

**Τέλος\_αν**

**Αν** count = ...**(6)**... **τότε**

done ← ...**(7)**...

...**(8)**... ← i

**αλλιώς**

i ← ...**(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.

**Μονάδες 10**

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

**ΣΥΝΑΡΤΗΣΗ A(x, ψ): ΠΡΑΓΜΑΤΙΚΗ**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** χ

**ΧΑΡΑΚΤΗΡΕΣ:** ψ[10]

**ΔΙΑΔΙΚΑΣΙΑ B(x, ψ, ζ)**

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ:** ψ

**ΠΡΑΓΜΑΤΙΚΕΣ:** χ, ζ

Επίσης δίνεται το τμήμα δηλώσεων κύριου προγράμματος:

**ΜΕΤΑΒΛΗΤΕΣ A**

**ΚΕΡΑΙΕΣ:** κ, λ[10], μ

**ΠΡΑΓΜΑΤΙΚΕΣ:** π, ρ[10], γ

**ΧΑΡΑΚΤΗΡΕΣ:** θ[10], υ

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

1. π ← A(κ, υ)

2. ΚΑΛΕΣΕ A(μ, θ)

3. ΚΑΛΕΣΕ B(π, μ)

4. υ ← A(μ, θ)

5. κ ← B(π, μ, ρ[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. (μονάδες 5)

**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. (μονάδες 5)

**Μονάδες 10**

### 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 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.

**Μονάδες 2**

**Γ2.** Να διαβάξει 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». (μονάδες 3) Να γίνεται έλεγχος εγκυρότητας ώστε ο αριθμός των επισκέψεων να μην είναι αρνητικός. (μονάδες 2)

**Μονάδες 5**

**Γ3.** Να βρίσκει και να εμφανίζει τον τίτλο του βίντεο με τον μεγαλύτερο αριθμό επισκέψεων. Να θεωρήσετε ότι είναι μοναδικό.

**Μονάδες 4**

**Γ4.** Να υπολογίζει για καθεμία 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.

**Μονάδες 6**

**Γ5.** Να βρίσκει και να εμφανίζει το όνομα της κατηγορίας επισκεψιμότητας στην οποία καταχωρίστηκαν τα περισσότερα βίντεο. Να θεωρήσετε ότι είναι μοναδική.

**Μονάδες 3**

#### Σημείωση

Το πλήθος των βίντεο δεν είναι γνωστό.

### ΘΕΜΑ Δ

Στην 27<sup>η</sup> 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.

**Μονάδες 2**

**Δ2.** Να διαβάξει επαναληπτικά τα ονόματα των μαθητών και να τα καταχωρίζει στον Πίνακα **ON**[40]. (μονάδα 1) 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. (μονάδες 2)

**Μονάδες 3**

**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). (μονάδα 1)

Η βαθμολογία να καταχωρίζεται στην αντίστοιχη θέση του Πίνακα **BAΘ**[40,6] μόνο αν είναι μεγαλύτερη από τη βαθμολογία που είναι ήδη καταχωρισμένη. (μονάδες 2)

Για τον τερματισμό της εισαγωγής δεδομένων το πρόγραμμα να εμφανίζει το μήνυμα «Υπάρχει νέα λύση προβλήματος; ΝΑΙ / ΟΧΙ». Αν εισαχθεί η τιμή «ΟΧΙ», να τερματίζεται η εισαγωγή δεδομένων. (μονάδες 2)

**Μονάδες 5**

**Δ4.** Να υπολογίζει και να καταχωρίζει στον Πίνακα **ΣΒ**[40] τα αθροίσματα των βαθμολογιών κάθε μαθητή στα έξι προβλήματα. Για τον σκοπό αυτό να καλεί μόνο μια φορά το υποπρόγραμμα με όνομα **ΥΣΒ**. (μονάδα 1) Να αναπτύξετε το υποπρόγραμμα **ΥΣΒ** το οποίο να δέχεται ως είσοδο τον Πίνακα **ΒΑΘ**[40,6] και να επιστρέφει ως έξοδο συμπληρωμένο τον Πίνακα **ΣΒ**[40]. (μονάδες 4)

**Μονάδες 5**

**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.

**Μονάδες 5**

#### Σημειώσεις

α) Δεν απαιτούνται έλεγχοι εγκυρότητας τιμών.

β) Να θεωρήσετε ότι θα δοθεί τουλάχιστον μια λύση προβλήματος από έναν μαθητή.