

ΘΕΜΑ Α

A1. Στις παρακάτω ερωτήσεις να απαντήσετε με το γράμμα Σ αν η πρόταση είναι Σωστή ή το γράμμα Λ αν η πρόταση είναι Λανθασμένη

1. Με το παρακάτω τμήμα κώδικα διαβάζουμε έναν αριθμό από το πληκτρολόγιο και τον εισάγουμε στην πρώτη θέση ενός πίνακα Π[N] ο οποίος έχει ήδη N-1 στοιχεία αφού προηγουμένως έχουμε μετακινήσει τα στοιχεία του μια θέση πιο κάτω.

```
Διάβασε X
Για i από 2 μέχρι N
    Π[i] ← Π[i - 1]
Τέλος_επανάληψης
Π[1] ← X
```

2. Η ταξινόμηση με επιλογή και η ταξινόμηση ευθείας ανταλλαγής έχουν την ίδια πολυπλοκότητα $O(n \log n)$
3. Στις Δυναμικές Δομές Δεδομένων έχουμε τη δυνατότητα να εισάγουμε και να διαγράφουμε κόμβους κατά βούληση.
4. Το κύριο χαρακτηριστικό των Λιστών είναι ότι από έναν κόμβο δεν υπάρχει μόνο ένας επόμενος κόμβος αλλά περισσότεροι.
5. Όλες οι Δομές Δεδομένων μπορούν να θεωρηθούν ειδικές περιπτώσεις γράφων.
6. Όταν δυο αλγόριθμοι με τα ίδια δεδομένα εισόδου, στις ίδιες συνθήκες δίνουν ίδια αποτελέσματα και ο Α απαιτεί λιγότερους πόρους(χρόνο ή μνήμη) τότε ο Β είναι λιγότερο αποδοτικός από τον Α.
7. Ένα από τα πλεονεκτήματα του Δομημένου Προγραμματισμού είναι η άμεση μεταφορά του προγράμματος σε αλγόριθμο.
8. Στη ΓΛΩΣΣΑ ισχύει η καθολική εμβέλεια των μεταβλητών και των σταθερών.
9. Η μερικώς περιορισμένη εμβέλεια διευκολύνει την ανάπτυξη ενός προγράμματος για τον αρχάριο προγραμματιστή.
10. Όταν το πρόγραμμα καλεί ένα υποπρόγραμμα και το υποπρόγραμμα καλεί με τη σειρά του ένα άλλο υποπρόγραμμα τότε η μέγιστη τιμή του δείκτη top της Στοίβας Χρόνου Εκτέλεσης είναι 3.

(ΜΟΝΑΔΕΣ 10)

A2. Εκμεταλλευόμενοι το παρακάτω τμήμα κώδικα το οποίο υπολογίζει το άθροισμα των στοιχείων της κύριας και της δευτερεύουσας διαγωνίου ενός πίνακα Π[1000,1000] και το οποίο είναι πολυπλοκότητας $O(n^2)$, να το τροποποιήσετε κατάλληλα ώστε να υπολογίζει το ίδιο με κώδικα πολυπλοκότητας $O(n)$, δηλαδή μόνο με μια επανάληψη και χωρίς δομή επιλογής.

```
σ1 ← 0
Για κ από 1 μέχρι n
    Για λ από 1 μέχρι n
        Αν κ = λ τότε
            σ1 ← σ1 + Π[κ, λ]
Τέλος_αν
Τέλος_επανάληψης
Τέλος_επανάληψης
```

```

σ2 ← 0
Για κ από 1 μέχρι n
  Για λ από 1 μέχρι n
    Αν κ + λ = n + 1 τότε
      σ2 ← σ2 + Π[κ, λ]
  Τέλος_αν
Τέλος_επανάληψης
Τέλος_επανάληψης
σ ← σ1 + σ2
Αν n mod 2 = 1 τότε
  σ ← σ - Π[n div 2, n div 2]
! Το μεσαίο στοιχείο
τέλος_αν

```

(ΜΟΝΑΔΕΣ 5)

A3. Να αντιστοιχίσετε τις λειτουργίες που περιγράφονται στην πρώτη στήλη με τον αντίστοιχο κώδικα που περιγράφεται στη δεύτερη στήλη και έπειτα με την αντίστοιχη πολυπλοκότητα που περιγράφεται στην τρίτη στήλη. Πχ $1 \rightarrow A \rightarrow I$

Ορισμένα στοιχεία της τρίτης στήλης δεν αντιστοιχίζονται πουθενά ενώ άλλα αντιστοιχίζονται σε περισσότερα από ένα στοιχεία των άλλων στηλών. **Για τις κατηγορίες χρονικής πολυπλοκότητας που δεν αντιστοιχίζονται πουθενά, γράψτε τι γνωρίζετε.**

Λειτουργίες		Κώδικας		Πολυπλοκότητα	
1.	Ώθηση ενός στοιχείου σε Στοιβά	A.	Δεδομένα // n, table, key // done ← ψευδής position ← 0 i ← 1 Όσο (done = ψευδής) και (i ≤ n) επανάλαβε Αν (table[i] = key) τότε done ← αληθής position ← i Τέλος_αν i ← i + 1 Τέλος_επανάληψης Αν (done = αληθής) τότε Εκτύπωσε "Το στοιχείο ", key, " ευρέθη στη θέση ", position Αλλιώς Εκτύπωσε "Το στοιχείο ", key, " δεν ευρέθη στον δοθέντα πίνακα" Τέλος_αν	I	O(1)
2.	Εισαγωγή ενός στοιχείου σε Ουρά	B.	Για I από 1 μέχρι n α ← 1 β ← 1 Για κ από 1 μέχρι n Για λ από 1 μέχρι n Αν Π[κ,λ] > Π[α,β] τότε α ← κ β ← λ τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αν I=1 τότε x ← Π[α,β]	II	O(logn)

			Αλλιώς Αν $\Pi[a,b] > \chi$ τότε $x \leftarrow \Pi[a,b]$ Τέλος_αν Τέλος_αν Τέλος_επανάληψης Αποτελέσματα // x //		
3.	Συγχώνευση ταξινομημένων Πινάκων ώστε να προκύψει ταξινομημένος Πίνακας	Γ.	Δεδομένα // n, table // Για i από 1 μέχρι n $j \leftarrow i$ Για από i + 1 μέχρι n Αν $table[k] < table[j]$ τότε $j \leftarrow k$ Τέλος_αν Τέλος_επανάληψης $temp \leftarrow table[j]$ $table[j] \leftarrow table[i]$ $table[i] \leftarrow temp$ Τέλος_επανάληψης Αποτελέσματα // table //	III	O(n)
4.	Ταξινόμηση με επιλογή (Selection Sort)	Δ.	Αν $top < N$ τότε Διάβασε X $top \leftarrow top + 1$ $A[top] \leftarrow X$ Αλλιώς Γράψε "Υπερχείλιση" Τέλος_αν	IV	O(n+m)
5.	Εύρεση Μέγιστου σε πολλούς διαστάτους πίνακες	Ε.	Αν $rear < N$ τότε Διάβασε X $πίσω \leftarrow πίσω + 1$ $A[πίσω] \leftarrow X$ Αλλιώς Γράψε "Η ουρά είναι γεμάτη" Τέλος_αν	V	O(n³)
6.	Απόθεση ενός στοιχείου από Στοιβα	Ζ.	Αν $front \leq rear$ τότε Γράψε $A[front]$ $front \leftarrow front + 1$ Αλλιώς Γράψε "Η ουρά είναι άδεια" Τέλος_αν	VI	O(2ⁿ)
7.	Εξαγωγή ενός Στοιχείου από Ουρά	Η.	Αν $top > 0$ τότε Γράψε $A[top]$ $top \leftarrow top - 1$ Αλλιώς Γράψε "Υποχείλιση" Τέλος_αν	VII	O(n²)
8.	Αναζήτηση (Σειριακή)	Θ.	Δεδομένα // n, m, x, y // $i \leftarrow 1$ $j \leftarrow 1$ $k \leftarrow 1$ Όσο $i \leq n$ και $j \leq m$ επανάλαβε Αν $x[i] < y[j]$ τότε $z[k] \leftarrow x[i]$ $i \leftarrow i + 1$ αλλιώς		

		<pre> z[k] <- y[j] j <- j + 1 Τέλος_αν k <- k + 1 Τέλος_επανάληψης Αν i > n τότε Για t από j μέχρι m z[k] <- y[t] k <- k + 1 Τέλος_επανάληψης αλλιώς Για t από i μέχρι n z[k] <- x[t] k <- k + 1 Τέλος_επανάληψης Τέλος_αν Αποτελέσματα // z //</pre>		
--	--	---	--	--

(ΜΟΝΑΔΕΣ 15)

A4. Να γράψετε τον κώδικα σε ΓΛΩΣΣΑ που να υλοποιεί τις παρακάτω σειρές – ακολουθίες – αθροίσματα:

1. $A = 1*2*3 + 2*3*4 + 4*5*6 ++v*(v+1)*(v+2)$

2. $\Gamma = 1 - 2^1 + 3^3! - 4^4! \pm v^v!$ όπου $v! = 1*2*3*.....v$

3. $\Delta = 1 + 2 + 3 + 5 + 8 + 13,$ για τους 100 πρώτους όρους. Αποτελεί την ακολουθία Fibonacci όπου ο πρώτος όρος της ακολουθίας είναι το 1, ο δεύτερος είναι το 2 και κάθε επόμενος όρος συντάσσεται από το άθροισμα των 2 αμέσως προηγούμενων του.

Να αναφέρετε για κάθε ένα τμήμα κώδικα που γράψατε τη χρονική πολυπλοκότητά του.

(ΜΟΝΑΔΕΣ 6)

A5. 1. Να γράψετε το αποτέλεσμα του παρακάτω τμήματος κώδικα για τον τριψήφιο θετικό αριθμό 345.

```

Διάβασε Χ
σ ← 0
Για i από 1 μέχρι 3
  σ ← σ*10 + Χ mod 10
  Χ ← Χ div 10
Τέλος_επανάληψης
Γράψε σ
```

2. Να επεκτείνετε τον κώδικα για αγνώστου πλήθους ψηφίων θετικών αριθμών. Να χρησιμοποιήσετε τη δομή επανάληψης Όσο...Επανάλαβε

(ΜΟΝΑΔΕΣ 4)

ΘΕΜΑ Β

Δίνεται το παρακάτω πρόγραμμα και 2 υποπρογράμματα.

```
ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ_Β
ΜΕΤΑΒΛΗΤΕΣ
  ΛΟΓΙΚΕΣ: FLAG
  ΑΚΕΡΑΙΕΣ: ΔΕΞΙ, ΑΡΙΣΤΕΡΟ, Χ, Ι
ΑΡΧΗ
  FLAG <- ΑΛΗΘΗΣ
  ΔΙΑΒΑΣΕ Χ
  ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 2
    ΑΝ ΟΧΙ FLAG ΤΟΤΕ
      ΔΕΞΙ <- Σ(Χ, FLAG) + 1
    ΑΛΛΙΩΣ
      ΑΡΙΣΤΕΡΟ <- Σ(Χ, FLAG) - 3 + 2*Χ
      FLAG <- ΟΧΙ (FLAG)
  ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΑΝ ΑΡΙΣΤΕΡΟ = ΔΕΞΙ ΤΟΤΕ
    ΓΡΑΨΕ "ΠΑΙΧΤΟΥΡΑ.ΤΟ ΒΡΗΚΕΣ!!!"
  ΑΛΛΙΩΣ
    ΓΡΑΨΕ "ΤΟΝ ΠΟΥΛΟ!"
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΣΥΝΑΡΤΗΣΗ Σ(Α, FLAG): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ

ΛΟΓΙΚΕΣ: FLAG

ΑΚΕΡΑΙΕΣ: Κ, Λ, Α

ΑΡΧΗ

ΑΝ FLAG = ΑΛΗΘΗΣ **ΤΟΤΕ**

Κ <- 2*Α - 1

Λ <- Α - 2

Σ <- ΠΟΛ_ΑΛΛΑ_ΡΩΣΙΚΑ(Κ, Λ)

ΑΛΛΙΩΣ

Κ <- Α

Λ <- 2 - Α

Σ <- ΠΟΛ_ΑΛΛΑ_ΡΩΣΙΚΑ(Κ, Λ)

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΣΥΝΑΡΤΗΣΗ ΠΟΛ_ΑΛΛΑ_ΡΩΣΙΚΑ(Α, Β): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α, Β, ΑΘΡ

ΑΡΧΗ

ΑΘΡ <- 0

ΟΣΟ Β > 0 **ΕΠΑΝΑΛΑΒΕ**

ΑΝ Β mod 2 = 1 **ΤΟΤΕ**

ΑΘΡ <- ΑΘΡ + Α

ΤΕΛΟΣ_ΑΝ

Α <- Α * 2

Β <- Β div 2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΠΟΛ_ΑΛΛΑ_ΡΩΣΙΚΑ <- ΑΘΡ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

B1. Να ξαναγράψετε το Κυρίως Πρόγραμμα που σας δίνεται, **απαλλαγμένο** από δομές επιλογής και επανάληψης ώστε να εκτελεί τις ίδιες λειτουργίες με το αρχικό. Διατηρείστε μόνο την τελευταία δομή επιλογής με τα μηνύματα και καμία άλλη.

(ΜΟΝΑΔΕΣ 5)

B2. Να μετατρέψετε τις 2 συναρτήσεις που σας δίνονται παραπάνω, σε μια διαδικασία (και μόνο μια!) η οποία να ενσωματώνει τις άλλες 2.

(ΜΟΝΑΔΕΣ 5)

B3. Να ξαναγράψετε το πρόγραμμα που προέκυψε από το B1 ερώτημα ώστε να επιτελεί τις ίδιες λειτουργίες, χωρίς τη χρήση υποπρογραμμάτων.

(ΜΟΝΑΔΕΣ 5)

B4. Να γράψετε στο τετράδιό σας τον ακέραιο θετικό αριθμό που θα πρέπει να δώσουμε στη μεταβλητή Χ αρχικά, ώστε να εμφανίζεται το μήνυμα «ΠΑΙΧΤΟΥΡΑ.ΤΟ ΒΡΗΚΕΣ!!!» στην οθόνη.

(ΜΟΝΑΔΕΣ 5)

(Για αποφυγή παρεξήγησης: Ο **Γεώργιος Πούλος** (Ρουμανία 1889- 11 Ιουνίου 1949) ήταν Έλληνας στρατιωτικός, αξιωματικός αρχικά του [Ελληνικού Στρατού](#) και κατά τη διάρκεια της [Γερμανική κατοχής της Ελλάδας](#) συνεργάτης των [Γερμανικών ναζιστικών δυνάμεων](#), ιδρυτής του τάγματος ασφαλείας [Ελληνικό Εθελοντικό Σώμα](#), γνωστό επίσης ως *Εθελοντικό Τάγμα Πούλου* ή *Poulos Verband* και υπεύθυνος για εγκλήματα πολέμου έναντι Ελλήνων αμάχων.)

ΘΕΜΑ Γ

- A.** Να δημιουργήσετε τη Συνάρτηση ΠΑΝ η οποία δέχεται ως είσοδο έναν Θετικό Ακέραιο Αριθμό και να αποφανθεί αν είναι ΠΑΝΨΗΦΙΑΚΟΣ. Πανψηφιακοί λέγονται οι αριθμοί οι οποίοι περιέχουν τουλάχιστον μια φορά **κάθε ψηφίο από το 0-9** (Πχ ο αριθμός 1234567890 είναι πανψηφιακός). Η συνάρτηση θα επιστρέφει τη λογική τιμή ΑΛΗΘΗΣ αν ο αριθμός που δέχτηκε σαν είσοδο είναι πανψηφιακός ή την τιμή ΨΕΥΔΗΣ στην αντίθετη περίπτωση.

(ΜΟΝΑΔΕΣ 5)

- B.** Να γίνει πρόγραμμα το οποίο:

- 1.** Να διαβάζει επαναληπτικά, θετικούς αριθμούς από το πληκτρολόγιο. Δεν χρειάζεται έλεγχος εγκυρότητας.

(ΜΟΝΑΔΕΣ 2)

- 2.** Να καλεί για κάθε έναν αριθμό τη συνάρτηση ΠΑΝ και να υπολογίζει και να εμφανίζει τον μεγαλύτερο **πανψηφιακό** αριθμό. Για τους **ΜΗ Πανψηφιακούς** αριθμούς να υπολογίζει το άθροισμά τους.

(ΜΟΝΑΔΕΣ 5)

- 3.** Κάθε φορά που στο άθροισμα προστίθεται ένας Μη Πανψηφιακός αριθμός, το πρόγραμμα να ελέγχει αν υπάρχει κάποιος αριθμός το πολύ ίσος με το υπάρχων άθροισμα που να **διαιρείται ακριβώς με κάθε έναν από τους ακέραιους αριθμούς 1 – 9**.

(ΜΟΝΑΔΕΣ 5)

- 4.** Η επαναληπτική διαδικασία να τερματίζει όταν βρεθεί ένας αριθμός που να ικανοποιεί το κριτήριο του 3^{ου} ερωτήματος.

(ΜΟΝΑΔΕΣ 3)

ΘΕΜΑ Δ

Σε έναν διαγωνισμό darts, οι κριτές ορίζουν τις 6 διαφορετικές βαθμολογίες σε κάθε ομόκεντρο κύκλο ενός στόχου, πριν την έναρξη του αγώνα. Για παράδειγμα μια ακολουθία βαθμών μπορεί να είναι 3 , 5 , 8 , 12 , 15 , 20 με το 20 να είναι η κεντρική βαθμολογία.



Να γίνει πρόγραμμα το οποίο:

- 1.** Θα διαβάζει τη βαθμολογία που έχουν αποφασίσει οι κριτές πριν τον αγώνα σε πίνακα Β[6] ελέγχοντας ώστε η πρώτη βαθμολογία να είναι θετικός αριθμός και κάθε επόμενη βαθμολογία να είναι μεγαλύτερη από την προηγούμενη.

2. Για κάθε διαγωνιζόμενο από τους 100 του διαγωνισμού:
- Na διαβάξει τα ονόματά τους σε πίνακα ON[100].
 - Na διαβάξει σε πίνακα ΑΠ[5,20] τα αποτελέσματα που έφερε σε 5 γύρους ρίχνοντας 20 βολές στον κάθε γύρο πραγματοποιώντας έλεγχο ώστε οι τιμές των αποτελεσμάτων να υπάρχουν στον πίνακα Β. Αν κάποια τιμή δεν υπάρχει τότε ο έλεγχος εγκυρότητας να την ξανά διαβάξει από την αρχή. Επιπλέον αν κάποιο βελάκι δεν βρει καθόλου στόχο, να επιτρέπει ο έλεγχος εγκυρότητας να εισαχθεί η τιμή -1 στην αντίστοιχη θέση. (Κάθε παίκτης έχει το δικό του πίνακα ΑΠ).
3. Na υπολογίζει και να αποθηκεύει σε πίνακα Ε[100], με μια θέση για κάθε διαγωνιζόμενο, τον αριθμό από τα βελάκια που δεν έχουν βρει στόχο και σε πίνακα Σ[100] τη συνολική βαθμολογία του κάθε διαγωνιζόμενου. Η συνολική βαθμολογία δίνεται από το άθροισμα των επιμέρους βαθμών του εκτός αυτών που δεν έχουν βρει στόχο και ως συνέπεια έχουν τιμή -1.
4. Na αποθηκεύει σε πίνακα ΣΥΧΝ[100,6] τις συχνότητες των βαθμολογιών που πέτυχε ο κάθε διαγωνιζόμενος.
5. Na εμφανίζει τα ονόματα των 3 πρώτων διαγωνιζομένων που είχαν τα περισσότερα βελάκια **εκτός** στόχου. (Δεν υπάρχουν ισοβαθμίες). Η απάντηση στην ερώτηση θα δοθεί καλώντας τη διαδικασία ΤΑΞΙΝΟΜΗΣΗ η οποία θα δέχεται τρεις πίνακες, έναν χαρακτήρων, έναν ακεραίων μονοδιάστατο και έναν ακεραίων δισδιάστατο, καθώς επίσης και ένα ακέραιο νούμερο που θα υποδηλώνει τον αριθμό των στοιχείων που θα πρέπει να εμφανιστούν. Εδώ αυτός ο αριθμός είναι 3 γιατί θέλουμε τους 3 πρώτους. Η ταξινόμηση θα κάνει χρήση του ακεραίου αριθμού κατά τέτοιο τρόπο ώστε να επιτυγχάνονται οι λιγότερες δυνατές επαναλήψεις, μόνο αυτές που είναι απαραίτητες.
6. Na εμφανίζει τα ονόματα των 10 πρώτων διαγωνιζομένων που είχαν το μεγαλύτερο σύνολο βαθμολογίας. Για το σκοπό αυτό να καλείται και πάλι η διαδικασία ΤΑΞΙΝΟΜΗΣΗ. (Προσοχή στα ονόματα λόγω της προηγούμενης ταξινόμησης και στα στοιχεία του δισδιάστατου πίνακα. Χρειαζόμαστε αντίγραφα!!)
7. (ΕΧΤΡΑΔΑΚΙ) Na διαβάξει τη συνολική βαθμολογία ενός διαγωνιζόμενου και να το αναζητά στον πίνακα Σ με τη βοήθεια της Συνάρτησης ΔΙΑΔ_ΑΝΑΖ η οποία θα πραγματοποιεί Δυναμική αναζήτηση στις 10 ταξινομημένες πρώτες θέσεις του πίνακα Σ[100]. Η συνάρτηση θα πρέπει να επιστρέφει τη θέση της προς αναζήτηση βαθμολογίας αν αυτή βρεθεί ή η τιμή 0 αν αυτή δεν βρεθεί. Σε περίπτωση που δεν υπάρχει τέτοια βαθμολογία, θα πρέπει να διαβάζεται εκ νέου μια βαθμολογία και να ξανακαλείται η παραπάνω συνάρτηση μέχρι η αναζήτηση να είναι επιτυχής. Τέλος να εμφανίζεται το όνομα του διαγωνιζόμενου που την είχε καθώς επίσης και η αναλυτική του βαθμολογία με την κάθε βαθμολογία και δίπλα τη συχνότητα εμφάνισής της. Τέλος να γραφεί η Συνάρτηση ΔΙΑΔ_ΑΝΑΖ.
8. Na γραφεί η Διαδικασία ΤΑΞΙΝΟΜΗΣΗ η οποία θα ταξινομεί σε αύξουσα σειρά τα στοιχεία του μονοδιάστατου πίνακα ακεραίων που αναφέραμε παραπάνω και θα παρασέρνει κατά την ταξινόμηση, τα στοιχεία του μονοδιάστατου και τα στοιχεία του δισδιάστατου πίνακα κατά γραμμές.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ

ΔΙΝΕΤΑΙ: ΣΥΝΤΕΛΕΣΤΗΣ ΦΠΑ: 24%