

ΠΛΗΡΟΦΟΡΙΚΗ
ΑΠΑΝΤΗΣΕΙΣ**ΘΕΜΑ Α****A1.**

1. ΛΑΘΟΣ
2. ΣΩΣΤΟ
3. ΣΩΣΤΟ
4. ΛΑΘΟΣ
5. ΣΩΣΤΟ

A2.**(α)**

- Υπολογισμός αθροισμάτων στοιχείων του πίνακα
- Εύρεση μεγίστου/ελαχίστου
- Ταξινόμηση των στοιχείων
- Αναζήτηση ενός στοιχείου
- Συγχώνευση δύο πινάκων

(β) Σχολικό Βιβλίο «Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον» σελ. 182
«ο αριθμός των πραγματικών ... του ίδιου τύπου».

(γ) HM(x), ΣΥΝ(x), ΕΦ(x), Ε(x), ΛΟΓ(x), T_P(x), A_T(x), A_M(x)

A3.**(α)**

i) 3 απωθήσεις

ii) Ο δείκτης top έχει την τιμή 3. Για να αδειάσει η στοίβα θα πρέπει ο δείκτης top να μηδενιστεί. Άρα μετά από μία απώθηση το top θα γίνει 2, μετά τη δεύτερη απώθηση θα γίνει 1 και μετά την τρίτη απώθηση ο δείκτης top θα γίνει 0 και θα έχει αδειάσει η στοίβα. Τα στοιχεία που βρίσκονται στις θέσεις 4 και 5 δεν μας απασχολούν διότι έχουν ήδη απωθηθεί από τη στοίβα και περιέχονται απλά στον πίνακα με τον οποίο υλοποιείται η στοίβα.

(β)

i) 2 εξαγωγές

ii) Οι δείκτες front και rear έχουν αντίστοιχα τις τιμές 3 και 4. Για να αδειάσει η ουρά θα πρέπει ο δείκτης front να πάρει τιμή μεγαλύτερη από αυτή του rear. Ομοίως με το προηγούμενο ερώτημα δεν μας απασχολούν τα στοιχεία στις θέσεις 1 και 2 διότι έχουν ήδη εξαχθεί από την ουρά και περιέχονται απλά στον πίνακα με τον οποίο υλοποιείται η ουρά. Οπότε, μετά από μία εξαγωγή ο δείκτης front θα πάρει την τιμή 4 και μετά τη δεύτερη εξαγωγή θα πάρει την τιμή 5, η οποία είναι μεγαλύτερη από την τιμή του rear.

A4.**(α)**

- i) 3 φορές
- ii) καμία φορά
- iii) 1 φορά

(β)

A+8 ή A+9

ΘΕΜΑ Β
B1.

```

ΑΝ Χ = 7 ΤΟΤΕ
  ΓΡΑΨΕ 'Α'
ΑΛΛΙΩΣ_ΑΝ Χ = 11 Η Χ = 13 ΤΟΤΕ
  ΓΡΑΨΕ 'Β'
ΑΛΛΙΩΣ_ΑΝ Χ < 20 ΤΟΤΕ
  ΓΡΑΨΕ 'Γ'
ΑΛΛΙΩΣ_ΑΝ Χ >= 50 ΚΑΙ Χ <= 100 ΤΟΤΕ
  ΓΡΑΨΕ 'Δ'
ΑΛΛΙΩΣ
  ΓΡΑΨΕ 'Ε'
ΤΕΛΟΣ_ΑΝ
  
```

B2.

- 1) ΑΛΗΘΗΣ
- 2) 2
- 3) $n \text{ MOD } i$
- 4) ΨΕΥΔΗΣ ***
- 5) ΟΧΙ(ΠΡΩΤΟΣ) (επίσης δεκτό και το ΠΡΩΤΟΣ=ΨΕΥΔΗΣ)

*** Το τμήμα προγράμματος που δίνεται δεν θα λειτουργήσει στην περίπτωση που δοθεί $n=2$. Για να επιτύχουμε το αποτέλεσμα και σε αυτήν την περίπτωση θα πρέπει η μεταβλητή ΠΡΩΤΟΣ να μένει αληθής παρόλο που για $n=2$ και $i=2$, θα εκτελεστεί η εντολή στην οποία βρίσκεται το 4^ο κενό. Άρα θέλουμε μια λογική έκφραση η οποία να γίνεται αληθής για $n=2$ και ψευδής για οποιαδήποτε άλλη τιμή. Οπότε η πιο ορθή αλγοριθμικά λύση είναι το $n=2$ ως λογική έκφραση με αποτέλεσμα λογική τιμή (αντί του ΨΕΥΔΗΣ). Το πρόβλημα αυτό όμως δημιουργείται καθώς έχει χρησιμοποιηθεί η Δομή Επανάληψης ΜΕΧΡΙΣ_ΟΤΟΥ, αντί της αναμφίβολα πιο κατάλληλης ΟΣΟ. Κάτι που συνεπάγεται ότι πιθανότατα πρόκειται περί αμέλειας στην εκφώνηση, παρά περί επιδίωξης τέτοιας επίλυσης.

ΘΕΜΑ Γ

ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ_Γ_ΠΑΝΕΛ_2020
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Όριο, φορτωμένα, πλήθος_δεν_φορτ, πλήθος_Ανω_1000, Υπόλοιπο, Βάρος

ΠΡΑΓΜΑΤΙΚΕΣ: Σύνολο, Κόστος

ΧΑΡΑΚΤΗΡΕΣ: Απάντηση

ΑΡΧΗ

```

ΓΡΑΨΕ 'Δώσε το όριο βάρους του πλοίου'
ΔΙΑΒΑΣΕ Όριο
ΓΡΑΨΕ 'Δώσε το βάρος των δεμάτων που είναι ήδη φορτωμένα'
ΔΙΑΒΑΣΕ φορτωμένα
ΟΣΟ φορτωμένα >= Όριο ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ 'λάθος εισαγωγή'
  ΔΙΑΒΑΣΕ φορτωμένα
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
πλήθος_δεν_φορτ <- 0
Σύνολο <- 0
πλήθος_Ανω_1000 <- 0
Υπόλοιπο <- Όριο - φορτωμένα
ΓΡΑΨΕ 'Μπορούν να φορτωθούν ακόμα ', Υπόλοιπο, ' κιλά.'
ΓΡΑΨΕ 'ΝΑ ΦΟΡΤΩΘΕΙ ΔΕΜΑ; (ΝΑΙ/ΟΧΙ)'
ΔΙΑΒΑΣΕ Απάντηση
ΟΣΟ Απάντηση = 'ΝΑΙ' ΕΠΑΝΑΛΑΒΕ
  ΓΡΑΨΕ 'Δώσε βάρος δέματος'
  ΔΙΑΒΑΣΕ Βάρος
  
```

```
ΑΝ Βάρος > Υπόλοιπο ΤΟΤΕ
  ΓΡΑΨΕ 'Το δέμα δεν χωράει'
  πλήθος_Δεν_φορτ <- πλήθος_Δεν_φορτ + 1
ΑΛΛΙΩΣ
ΑΝ Βάρος > 1000 ΤΟΤΕ
  πλήθος_Ανω_1000 <- πλήθος_Ανω_1000 + 1
ΤΕΛΟΣ_ΑΝ
ΑΝ Βάρος <= 500 ΤΟΤΕ
  Κόστος <- Βάρος*0.5
ΑΛΛΙΩΣ_ΑΝ Βάρος <= 1500 ΤΟΤΕ
  Κόστος <- 500*0.5 + (Βάρος - 500)*0.3
ΑΛΛΙΩΣ
  Κόστος <- 500*0.5 + 1000*0.3 + (Βάρος - 1500)*0.1
ΤΕΛΟΣ_ΑΝ
Σύνολο <- Σύνολο + Κόστος
φορτωμένα <- φορτωμένα + Βάρος
ΤΕΛΟΣ_ΑΝ
Υπόλοιπο <- Όριο - φορτωμένα
ΓΡΑΨΕ 'Μπορούν να φορτωθούν ακόμα ', Υπόλοιπο, ' κιλά.'
ΓΡΑΨΕ 'ΝΑ ΦΟΡΤΩΘΕΙ ΔΕΜΑ; (ΝΑΙ/ΟΧΙ) '
ΔΙΑΒΑΣΕ Απάντηση
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Δεν φορτώθηκαν ', πλήθος_Δεν_φορτ, ' δέματα'
ΓΡΑΨΕ 'Εισπράχθηκε το συνολικό ποσό των ', Σύνολο, '€'
ΓΡΑΨΕ 'φορτώθηκαν ', πλήθος_Ανω_1000, ' δέματα με βάρος πάνω από 1000'
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΘΕΜΑ Δ

ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ_Δ_ΠΑΝΕΛ_2020

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: *i*, *j*, *k*, ΠΛ, *max*, θετικά[20]

ΧΑΡΑΚΤΗΡΕΣ: Π[20], ΑΠ[20, 100], ΑΠΟΤ

ΑΡΧΗ

!Ερώτημα Δ2

ΓΙΑ *i* ΑΠΟ 1 ΜΕΧΡΙ 20

ΓΡΑΨΕ 'Δώσε όνομα περιοχής'

ΔΙΑΒΑΣΕ Π[*i*]

j <- 1

ΓΡΑΨΕ 'Δώσε αποτέλεσμα'

ΔΙΑΒΑΣΕ ΑΠΟΤ

ΟΣΟ ΑΠΟΤ <> 'ΤΕΛΟΣ' ΚΑΙ *j* <= 100 ΕΠΑΝΑΛΑΒΕ

ΑΠ[*i*, *j*] <- ΑΠΟΤ

j <- *j* + 1

ΓΡΑΨΕ 'Δώσε αποτέλεσμα'

ΔΙΑΒΑΣΕ ΑΠΟΤ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ *j* <= 100 ΤΟΤΕ

ΓΙΑ *k* ΑΠΟ *j* ΜΕΧΡΙ 100

ΑΠ[*i*, *k*] <- 'x'

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

!Ερώτημα Δ3

ΓΙΑ *i* ΑΠΟ 1 ΜΕΧΡΙ 20

ΠΛ <- 0

ΓΙΑ *j* ΑΠΟ 1 ΜΕΧΡΙ 100

ΑΝ ΑΠ[*i*, *j*] = 'θ' ΤΟΤΕ

ΠΛ <- ΠΛ + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

θετικά[*i*] <- ΠΛ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

max <- θετλκά[1]
ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 20
  ΑΝ θετλκά[i] > max ΤΟΤΕ
    max <- θετλκά[i]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'Περιοχές με τα περισσότερα θετικά δείγματα'
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  ΑΝ θετλκά[i] = max ΤΟΤΕ
    ΓΡΑΨΕ π[i]
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Το παραπάνω ερώτημα μπορεί να λυθεί και χωρίς max, μετά την ταξινόμηση
!Ερώτημα Δ4
ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(π, θετλκά)
ΓΡΑΨΕ 'Οι περιοχές ταξινομημένες'
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 20
  ΓΡΑΨΕ π[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΤΑΞΙΝΟΜΗΣΗ(π, θετλκά)
ΜΕΤΑΒΛΗΤΕΣ
  ΑΚΕΡΑΙΕΣ: i, j, θετλκά[20], TEMP
  ΧΑΡΑΚΤΗΡΕΣ: π[20], TEMP2
ΑΡΧΗ
  ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 20
    ΓΙΑ j ΑΠΟ 20 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1
      ΑΝ θετλκά[j] > θετλκά[j - 1] ΤΟΤΕ
        TEMP <- θετλκά[j]
        θετλκά[j] <- θετλκά[j - 1]
        θετλκά[j - 1] <- TEMP
        TEMP2 <- π[j]
        π[j] <- π[j - 1]
        π[j - 1] <- TEMP2
      ΑΛΛΙΩΣ
        ΑΝ θετλκά[j] = θετλκά[j - 1] ΤΟΤΕ
          ΑΝ π[j] < π[j - 1] ΤΟΤΕ
            TEMP2 <- π[j]
            π[j] <- π[j - 1]
            π[j - 1] <- TEMP2
          ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
      ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
    
```

Οι παραπάνω λύσεις είναι ενδεικτικές