

ΘΕΜΑ Α

A1. 1. ΛΑΘΟΣ 2. ΣΩΣΤΟ 3. ΛΑΘΟΣ 4. ΣΩΣΤΟ 5. ΛΑΘΟΣ

A2.

1. Το προγραμματιστικό περιβάλλον εντοπίζει τα λάθη που εμφανίζονται κατά τον σχεδιασμό (συντακτικά λάθη) και κατά την εκτέλεση. Τα λάθη που δεν εντοπίζει το προγραμματιστικό περιβάλλον είναι τα λογικά.

ΣΥΝΤΑΚΤΙΚΟ ΛΑΘΟΣ:

ΚΛΕΣΕ ΔΙΑΔ(παραμετρος1, παραμετρος2)

ΛΑΘΟΣ ΚΑΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ:

μεταβλητη1 \leftarrow μεταβλητη2 / 0

ΛΟΓΙΚΟ ΛΑΘΟΣ:

Αν θέλουμε να υπολογίσουμε το παραγοντικό του 5 στη ΓΛΩΣΣΑ θα πρέπει να ακολουθηθεί το παρακάτω τμήμα προγράμματος:

παραγοντικό \leftarrow 1

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 5

παραγοντικό \leftarrow παραγοντικό * Ι

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Λογικό λάθος θα είχαμε αν είχε αρχικοποιηθεί η μεταβλητή 'παραγοντικό' με την τιμή 0.

2. Ορισμός: Δοθέντων των στοιχείων a_1, a_2, \dots, a_n η ταξινόμηση συνίσταται στη μετάθεση (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά $a_{k1}, a_{k2}, \dots, a_{kn}$ έτσι ώστε, δοθείσης μίας συνάρτησης διάταξης (ordering function), f , να ισχύει: $f(a_{k1}) \leq f(a_{k2}) \leq \dots \leq f(a_{kn})$.

Σκοπός της ταξινόμησης είναι η ευκολότερη αναζήτηση ενός στοιχείου.

3. Όταν μια τιμή προκύπτει από υπολογισμό, τότε αναφερόμαστε σε εκφράσεις. Για τη σύνταξη μιας αριθμητικής έκφρασης χρησιμοποιούνται αριθμητικές σταθερές, μεταβλητές, συναρτήσεις, αριθμητικοί τελεστές και παρενθέσεις. Οι αριθμητικές εκφράσεις υλοποιούν απλές ή σύνθετες μαθηματικές πράξεις. Κάθε έκφραση παριστάνει μια συγκεκριμένη αριθμητική τιμή, η οποία βρίσκεται μετά την εκτέλεση των πράξεων. Γι' αυτό είναι απαραίτητο όλες οι μεταβλητές, που εμφανίζονται σε μια έκφραση να έχουν οριστεί προηγουμένως, δηλαδή, να έχουν κάποια τιμή. Η εκτέλεση των πράξεων εξαρτάται από την ιεραρχία των πράξεων.

A3. 1. Β) 2. Γ) 3. Β)

- A4. Η δομή δεδομένων που πρέπει να χρησιμοποιηθεί για την υλοποίηση της σειράς των μαθητών είναι η ΟΥΡΑ. Αυτό γιατί, πρέπει να διατηρηθεί η μέθοδος FIFO (Πρώτος Μέσα, Πρώτος Έξω) ώστε ο μαθητής ο οποίος θα σταθεί πρώτος στην έδρα να εξυπηρετηθεί και να φύγει πρώτος, και ο μαθητής ο οποίος θα σταθεί τελευταίος να φύγει τελευταίος.

ΘΕΜΑ Β

- B1. 1) K 2) I 3) A 4) B 5) ΠΡΩΤΟΙ_ΚΟΙΝΟΙ_B[I] 6) ΠΡΩΤΟΙ_ΚΟΙΝΟΙ_B[I] 7) B / I
8) 0 9) I ^ ΠΡΩΤΟΙ_ΚΟΙΝΟΙ_B[I] 10) I ^ ΠΡΩΤΟΙ_ΚΟΙΝΟΙ_A[I] 11) 1
12) ΜΚΔ

- B2. 1.

```
ΣΥΝΑΡΤΗΣΗ ΨΗΦΙΑ(αριθμος):ΑΚΕΡΑΙΑ
ΜΕΤΑΛΒΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: αριθμος, ανω, κατω
ΑΡΧΗ
    ανω ← 9
    κατω ← 1
    αριθμος ← A_T(αριθμος)
    ΨΗΦΙΑ ← 1
    ΑΝ αριθμος <> 0 ΤΟΤΕ
        ΟΣΟ ΟΧΙ(αριθμος >= κατω ΚΑΙ αριθμος <= ανω) ΕΠΑΝΑΛΑΒΕ
            ανω ← ανω*10 + 9
            κατω ← κατω*10
            ΨΗΦΙΑ ← ΨΗΦΙΑ + 1
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

Ο μαθητής μπορεί κάλλιστα να υλοποιήσει τον αλγόριθμο με διαδικασία και θα βαθμολογηθεί ως σωστό.

2. Ο καταλληλότερος τύπος υποπρογράμματος για την υλοποίηση του παραπάνω αλγόριθμου είναι η συνάρτηση, διότι επιστρέφεται μόνο μία τιμή, σε αντίθεση με την διαδικασία η οποία μπορεί να επιστρέφει πολλαπλές τιμές.

ΘΕΜΑ Γ

```

ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ_Γ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, Κ, ΔΥΝ, ΠΡΩΤ_Ψ, ΠΡΩΤΟ[6200000,10], ΠΛ_1_ΩΣ_9[9,10], ΠΛ
    ΑΚΕΡΑΙΕΣ: ΠΛ_ΔΙΑΦ, ΜΑΧ, ΜΑΧ_ΕΤΟΣ, ΔΙΑΦ_5[9] !Α' ΤΡΟΠΟΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΕΙΣ[6200000,10], ΣΥΧΝ[9,10]
    ΛΟΓΙΚΕΣ: ΔΙΑΦ_5[9] !Β' ΤΡΟΠΟΣ
ΑΡΧΗ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6200000
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10
        ΔΙΑΒΑΣΕ ΕΙΣ[Ι,Κ] !ΕΡ Γ2
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6200000
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10
        ΔΥΝ <- 0 !ΕΡ Γ3
        ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
            ΔΥΝ <- ΔΥΝ + 1
            ΑΝ ΕΙΣ[Ι,Κ] < 10 ΤΟΤΕ
                ΠΡΩΤ_Ψ <- Α_Μ(ΕΙΣ[Ι,Κ])
            ΑΛΛΙΩΣ
                ΠΡΩΤ_Ψ <- Α_Μ(ΕΙΣ[Ι,Κ]/10^ΔΥΝ)
            ΤΕΛΟΣ_ΑΝ
            ΜΕΧΡΙΣ_ΟΤΟΥ ΠΡΩΤ_Ψ < 10
            ΠΡΩΤΟ[Ι,Κ] <- ΠΡΩΤ_Ψ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 9
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10
        ΠΛ_1_ΩΣ_9[Ι,Κ] <- 0 !ΕΡ Γ4
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΠΛ <- 0
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 6200000
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10
        ΑΝ ΠΡΩΤΟ[Ι,Κ] > 0 ΤΟΤΕ
            ΠΛ_1_ΩΣ_9[ΠΡΩΤΟ[Ι,Κ],Κ] <- ΠΛ_1_ΩΣ_9[ΠΡΩΤΟ[Ι,Κ],Κ] + 1
            ΠΛ <- ΠΛ + 1
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 9
    ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 10
        ΣΥΧΝ[Ι,Κ] <- ΠΛ_1_ΩΣ_9[Ι,Κ]/ΠΛ*100 !ΕΡ Γ4
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΜΑΧ <- -1
ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ 10
    ΚΑΛΕΣΕ ΕΛΕΓΧΟΣ(ΣΥΧΝ, Ι, ΔΙΑΦ_5) !ΕΡ Γ5
    !μπορούμε να εισάγουμε μετρητή ως παράμετρο επειδή δεν αλλάζει η
    !τιμή του μέσα στη διαδικασία ΕΛΕΓΧΟΣ
    ΠΛ_ΔΙΑΦ <- 0

```

```

!Α' ΤΡΟΠΟΣ
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 9
    ΠΛ_ΔΙΑΦ <- ΠΛ_ΔΙΑΦ + ΔΙΑΦ_5[Κ]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
!Β' ΤΡΟΠΟΣ
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 9
    ΑΝ ΔΙΑΦ_5[Κ] = ΨΕΥΔΗΣ ΤΟΤΕ
        ΠΛ_ΔΙΑΦ <- ΠΛ_ΔΙΑΦ + 1
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ ΠΛ_ΔΙΑΦ >= 2 ΤΟΤΕ
    ΓΡΑΨΕ 'ΠΙΘΑΝΩΣ ΜΗ ΕΓΚΥΡΑ ΔΕΔΟΜΕΝΑ' , 2010 + Ι !ΕΡ Γ6
ΤΕΛΟΣ_ΑΝ
ΑΝ ΠΛ_ΔΙΑΦ > ΜΑΧ ΤΟΤΕ
    ΜΑΧ <- ΠΛ_ΔΙΑΦ !ΕΡ Γ7
    ΜΑΧ_ΕΤΟΣ <- 2010 + Ι
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΓΡΑΨΕ 'ΤΟ ΕΤΟΣ ΜΕ ΤΙΣ ΠΕΡΙΣΣΟΤΕΡΕΣ ΥΠΕΡΒΑΙΝΟΥΣΕΣ ΔΙΑΦΟΡΕΣ:' , ΜΑΧ_ΕΤΟΣ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΛΕΓΧΟΣ(ΣΥΧΝ, Ι, ΔΙΑΦ_5) !ΕΡ Γ8
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ι, Κ, ΔΙΑΦ_5[9] !Α' ΤΡΟΠΟΣ
    ΛΟΓΙΚΕΣ: ΔΙΑΦ_5[9] !Β' ΤΡΟΠΟΣ
    ΠΡΑΓΜΑΤΙΚΕΣ: ΜΠΕΝΦ[9], ΣΥΧΝ[9,10]

ΑΡΧΗ
ΜΠΕΝΦ[1] <- 30.1
ΜΠΕΝΦ[2] <- 17.6
ΜΠΕΝΦ[3] <- 12.5
ΜΠΕΝΦ[4] <- 9.7
ΜΠΕΝΦ[5] <- 7.9
ΜΠΕΝΦ[6] <- 6.7
ΜΠΕΝΦ[7] <- 5.8
ΜΠΕΝΦ[8] <- 5.1
ΜΠΕΝΦ[9] <- 4.6
ΓΙΑ Κ ΑΠΟ 1 ΜΕΧΡΙ 9
    ΑΝ Α_Τ(ΣΥΧΝ[Κ, Ι] - ΜΠΕΝΦ[Κ])/ΜΠΕΝΦ[Κ]*100 <= 5 ΤΟΤΕ
        ΔΙΑΦ_5[Κ] <- 0 !Α' ΤΡΟΠΟΣ
        ΔΙΑΦ_5[Κ] <- ΑΛΗΘΗΣ !Β' ΤΡΟΠΟΣ
    ΑΛΛΙΩΣ
        ΔΙΑΦ_5[Κ] <- 1 !Α' ΤΡΟΠΟΣ
        ΔΙΑΦ_5[Κ] <- ΨΕΥΔΗΣ !Β' ΤΡΟΠΟΣ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

ΘΕΜΑ Δ

ΠΡΟΓΡΑΜΜΑ ΘΕΜΑ_Δ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: N1, ΚΟΛΑΤΖ[1000], ΧΙΛΙΑΔΑ, A, MAX_APT, MAX_PEP, MAX_APT_B

ΑΚΕΡΑΙΕΣ: MAX_PEP_B, I, ΒΗΜΑ

ΠΡΑΓΜΑΤΙΚΕΣ: N

ΛΟΓΙΚΕΣ: FLAG

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ N !ΕΡ Δ2

ΜΕΧΡΙΣ_ΟΤΟΥ N>0 ΚΑΙ N=A_M(N)

N1 <- N

ΧΙΛΙΑΔΑ <- 0

A <- 1

ΓΙΑ I ΑΠΟ 1 ΜΕΧΡΙ 1000

ΚΟΛΑΤΖ[I] <- 0

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΚΟΛΑΤΖ[1] <- N1

MAX_APT <- -1

MAX_PEP <- -1

FLAG <- ΨΕΥΔΗΣ

ΟΣΟ ΚΟΛΑΤΖ[A] <> 1 ΚΑΙ FLAG = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ !ΕΡ Δ3

ΑΝ A = 1000 ΤΟΤΕ

A <- 0

ΧΙΛΙΑΔΑ <- ΧΙΛΙΑΔΑ + 1

ΤΕΛΟΣ_ΑΝ

ΑΝ N1 MOD 2 = 0 ΤΟΤΕ

ΑΝ N1 > MAX_APT ΤΟΤΕ !ΕΡ Δ6

MAX_APT <- N1

MAX_APT_B <- ΧΙΛΙΑΔΑ*1000 + A

ΤΕΛΟΣ_ΑΝ

N1 <- N1/2

ΑΛΛΙΩΣ

ΑΝ N1 > MAX_PEP ΤΟΤΕ !ΕΡ Δ6

MAX_PEP <- N1

MAX_PEP_B <- ΧΙΛΙΑΔΑ*1000 + A

ΤΕΛΟΣ_ΑΝ

N1 <- N1*3 + 1

ΤΕΛΟΣ_ΑΝ

A <- A + 1

ΚΟΛΑΤΖ[A] <- N1

I <- 1

ΟΣΟ I <= 1000 ΚΑΙ FLAG = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ

ΑΝ I <> A ΤΟΤΕ

FLAG <- ΚΟΛΑΤΖ[I] = ΚΟΛΑΤΖ[A]

ΤΕΛΟΣ_ΑΝ

I <- I + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ FLAG ΤΟΤΕ !ΕΡ Δ4

ΒΡΕΘΗΚΕ <- ΨΕΥΔΗΣ

I <- 1

```

ΟΣΟ I <= 1000 ΚΑΙ ΒΡΕΘΗΚΕ = ΨΕΥΔΗΣ ΕΠΑΝΑΛΑΒΕ
  ΑΝ I <> A ΤΟΤΕ
    ΑΝ ΚΟΛΑΤΖ[I] = ΚΟΛΑΤΖ[A] ΚΑΙ I < A ΤΟΤΕ
      ΒΗΜΑ <- ΧΙΛΙΑΔΑ*1000 + I
      ΒΡΕΘΗΚΕ <- ΑΛΗΘΗΣ
    ΑΛΛΙΩΣ_ΑΝ ΚΟΛΑΤΖ[I] = ΚΟΛΑΤΖ[A] ΚΑΙ I > A ΤΟΤΕ
      ΒΗΜΑ <- (ΧΙΛΙΑΔΑ-1)*1000 + I
      ΒΡΕΘΗΚΕ <- ΑΛΗΘΗΣ
    ΤΕΛΟΣ_ΑΝ
  ΤΕΛΟΣ_ΑΝ
  I <- I + 1
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ
  ΒΗΜΑ <- ΧΙΛΙΑΔΑ*1000 + A
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ 'Α:', ΒΗΜΑ
ΑΝ FLAG ΤΟΤΕ
  ΓΡΑΨΕ N, ΚΟΛΑΤΖ[A] !ΕΡ Δ5
ΤΕΛΟΣ_ΑΝ
ΓΡΑΨΕ 'ΜΕΓΙΣΤΟΣ ΑΡΤΙΟΣ ΑΡΙΘΜΟΣ:', MAX_ΑΡΤ, 'ΜΕ ΒΗΜΑ:', MAX_ΑΡΤ_Β !ΕΡ Δ6
ΓΡΑΨΕ 'ΜΕΓΙΣΤΟΣ ΠΕΡΙΤΤΟΣ ΑΡΙΘΜΟΣ:', MAX_ΠΕΡ, 'ΜΕ ΒΗΜΑ:', MAX_ΠΕΡ_Β
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```