

**Επαναληπτική δοκιμασία στην
Ανάπτυξη Εφαρμογών σε Προγραμματιστικό
Περιβάλλον**

Τεχνολογικής Κατεύθυνσης

Μάιος 2013

ΘΕΜΑ Α

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

1. Η παραγοντοποίηση ενός θετικού ακεραίου αριθμού N , σε γινόμενο πρώτων παραγόντων, αποτελεί πρόβλημα βελτιστοποίησης.
2. Κάθε μεταβλητή παίρνει τιμή μόνο με την εντολή εκχώρησης.
3. Η χρήση πινάκων είναι ένας βολικός τρόπος για τη διαχείριση πολλών δεδομένων διαφορετικού τύπου.
4. Ο τρόπος σύνταξης μιας διαδικασίας διαφέρει απ' τον τρόπο σύνταξης μιας συνάρτησης, ενώ ο τρόπος κλήσης τους είναι ο ίδιος.
5. Όταν σ' ένα υποπρόγραμμα έχουμε παράμετρο πίνακα, τότε η τυπική και η αντίστοιχη πραγματική παράμετρος, πρέπει να είναι ίδιου τύπου και ίδιου μεγέθους.

Μονάδες 10

A2. α) Να γράψετε σε Γλώσσα την εντολή εκχώρησης που αντιστοιχεί στην παρακάτω αριθμητική παράσταση.

$$X_1 = \frac{-\beta + \sqrt{\Delta}}{2\alpha}$$

Μονάδες 3

- β) Να γράψετε σε ΓΛΩΣΣΑ τη λογική έκφραση (συνθήκη), η οποία ελέγχει τις τιμές δύο μεταβλητών x, ψ όπως περιγράφεται στην πρόταση:
το x είναι μικρότερο του 10 ή το ψ μικρότερο του 10, αλλά όχι και τα δύο συγχρόνως.

Μονάδες 3

A3. Να αντιστοιχίσετε κάθε τμήμα ψευδοκώδικα της Στήλης A με το ισοδύναμό του τμήμα προγράμματος της Στήλης B.

Γράψτε δηλαδή στο τετράδιό σας για κάθε αριθμό 1-3 της Στήλης A ένα γράμμα α-γ της Στήλης B.

| Στήλη A | Στήλη B |
|---|---|
| 1.10 ΔΙΑΒΑΣΕ A 20 AN A<=0 ΤΟΤΕ GOTO 50 30 ΓΡΑΨΕ 'ΘΕΤΙΚΟ' 40 GOTO 10 50 ΓΡΑΨΕ 'ΜΗ ΘΕΤΙΚΟ' 60 ΤΕΛΟΣ | α. ΔΙΑΒΑΣΕ A ΟΣΟ A<=0 ΕΠΑΝΑΛΑΒΕ ΓΡΑΨΕ 'ΜΗ ΘΕΤΙΚΟ' ΔΙΑΒΑΣΕ A ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ ΓΡΑΨΕ 'ΘΕΤΙΚΟ' ΤΕΛΟΣ |
| 2.10 ΔΙΑΒΑΣΕ A 20 AN A<=0 ΤΟΤΕ GOTO 50 30 ΓΡΑΨΕ 'ΘΕΤΙΚΟ' 40 GOTO 60 50 ΓΡΑΨΕ 'ΜΗ ΘΕΤΙΚΟ' 60 ΤΕΛΟΣ | β. ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ ΔΙΑΒΑΣΕ A AN A>0 ΤΟΤΕ ΓΡΑΨΕ 'ΘΕΤΙΚΟ' ΤΕΛΟΣ_ΑΝ ΜΕΧΡΙΣ_ΟΤΟΥ A<=0 ΓΡΑΨΕ 'ΜΗ ΘΕΤΙΚΟ' ΤΕΛΟΣ |
| 3.10 ΔΙΑΒΑΣΕ A 20 AN A>0 ΤΟΤΕ GOTO 50 30 ΓΡΑΨΕ 'ΜΗ ΘΕΤΙΚΟ' 40 GOTO 10 50 ΓΡΑΨΕ 'ΘΕΤΙΚΟ' 60 ΤΕΛΟΣ | γ. ΔΙΑΒΑΣΕ A AN A>0 ΤΟΤΕ ΓΡΑΨΕ 'ΘΕΤΙΚΟ' ΑΛΛΙΩΣ ΓΡΑΨΕ 'ΜΗ ΘΕΤΙΚΟ' ΤΕΛΟΣ_ΑΝ ΤΕΛΟΣ |

Μονάδες 9

A4. Να γράψετε αλγόριθμο που διαβάζει δύο αριθμούς και τυπώνει τον μεγαλύτερο, με τη χρήση μόνο μίας απλής εντολής επιλογής. Θεωρήστε δεδομένο ότι δίνονται δύο άνισοι αριθμοί.

Μονάδες 9

A5. Δίνεται η παρακάτω συνάρτηση η οποία υλοποιεί τον αλγόριθμο της σειριακής αναζήτησης ακέραιου αριθμού σε πίνακα ακεραίων, στον οποίο θεωρούμε ότι μπορεί να υπάρχει το πολύ μία φορά. Η αναζήτηση θα σταματήσει μόλις βρεθεί ο αριθμός ή όταν ελεγχθούν όλες οι θέσεις του πίνακα. Όταν βρεθεί ο αριθμός η συνάρτηση θα επιστρέφει την θέση του μέσα στον πίνακα, διαφορετικά την τιμή 0. Η συνάρτηση έχει τυπικές παραμέτρους πίνακα 1000 ακεραίων θέσεων και μια ακέραια μεταβλητή, την τιμή της οποίας θα αναζητήσει.

ΣΥΝΑΡΤΗΣΗ Sequential_Search(table, key): ΑΚΕΡΑΙΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: table[1000], key, i, position

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

done ← ΨΕΥΔΗΣ

position ← 0

i ← 1

ΌΣΟ (done = ΨΕΥΔΗΣ) ΚΑΙ (i ≤ 1000) ΕΠΑΝΑΛΑΒΕ

ΑΝ table[i] = key ΤΟΤΕ

done ← ΑΛΗΘΗΣ

position ← i

ΑΛΛΙΩΣ

i ← i + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Sequential_Search ← position

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Η παρακάτω Συνάρτηση αν συμπληρωθεί σωστά υλοποιεί την ίδια διεργασία με την προηγούμενη, στηριζόμενη όμως σε πιο αποδοτικό αλγόριθμο (ταχύτερη εκτέλεση).

ΣΥΝΑΡΤΗΣΗ Sequential_Search(table, key): ΑΚΕΡΑΙΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: table[1000], key, i

ΑΡΧΗ

$i \leftarrow \dots$

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

$i \leftarrow i+1$

ΜΕΧΡΙΣ_ΟΤΟΥ (...) Ή (i=1000)

ΑΝ ... ΤΟΤΕ

Sequential_Search $\leftarrow i$

ΑΛΛΙΩΣ

Sequential_Search $\leftarrow 0$

ΤΕΛΟΣ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Να γράψετε στο τετράδιο σας τις τρεις (3) γραμμές της συνάρτησης που έχουν κενά ..., κατάλληλα συμπληρωμένες.

Μονάδες 6

ΘΕΜΑ Β

B1. Δίνεται το παρακάτω πρόγραμμα το οποίο καλεί τα 3 υποπρογράμματα που ακολουθούν:

ΠΡΟΓΡΑΜΜΑ ΠΡΑΞΕΙΣ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, Γ, X, Y, P

ΑΡΧΗ

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ A, B

ΜΕΧΡΙΣ_ΟΤΟΥ (A>0) ΚΑΙ (B>0)

Γ $\leftarrow \Pi(A, B)$

ΚΑΛΕΣΕ Δ(A, B, X, Y)

P $\leftarrow \Delta N(A, B)$

ΓΡΑΨΕ Γ, X, Y, P

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ Π (Α1, Α2) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α1, Α2, Ι, Σ

ΑΡΧΗ

Σ ← 0

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Α1

Σ ← Σ + Α2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

Π ← Σ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΔΙΑΔΙΚΑΣΙΑ Δ (Α1, Α2, Π, Υ)
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α1, Α2, Π, Υ

ΑΡΧΗ

Π ← 0

Υ ← Α1

ΟΣΟ Υ ≥ Α2 ΕΠΑΝΑΛΑΒΕ

Π ← Π + 1

Υ ← Υ - Α2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΣΥΝΑΡΤΗΣΗ ΔΝ (Α1, Α2) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α1, Α2, Ι, Γ

ΑΡΧΗ

Γ ← 1

ΓΙΑ Ι ΑΠΟ 1 ΜΕΧΡΙ Α2

Γ ← Γ * Α1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΔΝ ← Γ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Να γράψετε στο τετράδιό σας, τις τέσσερις (4) εντολές εκχώρησης του παρακάτω προγράμματος που περιέχουν κενά, συμπληρωμένες στη θέση των

κενών με τους κατάλληλους αριθμητικούς τελεστές, έτσι ώστε το πρόγραμμα να δίνει ισοδύναμο αποτέλεσμα με το αρχικό.

```
ΠΡΟΓΡΑΜΜΑ ΠΡΑΞΕΙΣ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Α, Β, Γ, Χ, Υ, Ρ
ΑΡΧΗ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ Α, Β
ΜΕΧΡΙΣ_ΟΤΟΥ (Α>0) ΚΑΙ (Β>0)
    Γ←Α ... Β
    Χ←Α ... Β
    Υ←Α ... Β
    Ρ←Α ... Β
    ΓΡΑΨΕ Γ, Χ, Υ, Ρ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Μονάδες 12

B2. Δίνεται το παρακάτω πρόγραμμα το οποίο ελέγχει αν η συνάρτηση ΑΛΑ_ΡΩΣΙΚΑ, η οποία υλοποιεί τον αλγόριθμο του Ρώσικου πολλαπλασιασμού, υπολογίζει σωστά το γινόμενο 2 θετικών ακεραίων αριθμών.

```
ΠΡΟΓΡΑΜΜΑ ΕΛΕΓΧΟΣ
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Α, Β, Γ
ΑΡΧΗ
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΔΙΑΒΑΣΕ Α, Β
ΜΕΧΡΙΣ_ΟΤΟΥ (Α>0) ΚΑΙ (Β>0)
    Γ ← ΑΛΑ_ΡΩΣΙΚΑ(Α, Β)
    ΑΝ Α * Β = Γ ΤΟΤΕ
        ΓΡΑΨΕ 'ΣΩΣΤΟ ΓΙΝΟΜΕΝΟ'
    ΑΛΛΙΩΣ
        ΓΡΑΨΕ 'ΛΑΘΟΣ ΓΙΝΟΜΕΝΟ'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΣΥΝΑΡΤΗΣΗ ΑΛΛΑ_ΡΩΣΙΚΑ (A1, A2) : ΑΚΕΡΑΙΑ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A1, A2, Σ

ΑΡΧΗ

Σ ← 0

ΟΣΟ A2 > 0 ΕΠΑΝΑΛΑΒΕ

ΑΝ A2 MOD 2 = 1 ΤΟΤΕ

Σ ← Σ + A1

ΤΕΛΟΣ_ΑΝ

A1 ← A1 * 2

A2 ← A2 DIV 2

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΛΑ_ΡΩΣΙΚΑ ← Σ

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

α) Να γράψετε διαδικασία ισοδύναμη με τη συνάρτηση.

Μονάδες 4

β) Να ξαναγράψετε το πρόγραμμα ώστε να χρησιμοποιεί την διαδικασία.

Μονάδες 4

ΘΕΜΑ Γ

Το γνωστό παιγνίδι «ΚΡΕΜΑΛΑ» θα μπορούσε να υλοποιηθεί σε υπολογιστή ως εξής: Η άγνωστη λέξη θα εμφανίζεται στην οθόνη του υπολογιστή με τη γνωστή μορφή, δηλαδή θα εμφανίζονται το πρώτο και το τελευταίο γράμμα και ενδιάμεσα παύλες (π.χ. Α - - - - - Σ). Ο χρήστης-παίκτης θα προσπαθεί να τη βρει δίνοντας ένα-ένα τα γράμματα που πιστεύει ότι περιέχει. Σε κάθε προσπάθεια το πρόγραμμα θα επανεμφανίζει τη λέξη με τις παύλες, ενημερωμένη με τα γράμματα που πιθανόν βρέθηκαν. Το παιγνίδι θα τελειώνει

με νικητή το χρήστη όταν βρίσκει τη λέξη, δίνοντάς του τη δυνατότητα να δώσει μέχρι και 7 λανθασμένα γράμματα.

Να αναπτύξετε πρόγραμμα σε «ΓΛΩΣΣΑ» το οποίο :

Γ1. Περιλαμβάνει τμήμα δηλώσεων μεταβλητών.

μονάδες 2

Γ2. Καλεί προκατασκευασμένο υποπρόγραμμα απ' τη βιβλιοθήκη της «ΓΛΩΣΣΑΣ», το οποίο ονομάζεται ΔΗΜΙΟΥΡΓΙΑ_ΛΕΞΗΣ(P) και έχει τυπική παράμετρο πίνακα P[8] χαρακτήρων. Ο πίνακας που επιστρέφει το υποπρόγραμμα περιέχει σε κάθε μια απ' τις οκτώ θέσεις του ένα κεφαλαίο ελληνικό γράμμα, δηλαδή συνολικά μια λέξη οκτώ γραμμάτων. (Το υποπρόγραμμα αυτό, δεν θα το κατασκευάσετε, απλώς θα το καλέσετε μια φορά στην αρχή για να σας δημιουργήσει την άγνωστη λέξη, την οποία επιλέγει τυχαία από ένα αρχείο με πολλές λέξεις οκτώ γραμμάτων.)

μονάδα 1

Γ3. Πριν απ' το διάβασμα κάθε γράμματος που θα δίνει ο χρήστης, το πρόγραμμα θα καλεί προκατασκευασμένο υποπρόγραμμα απ' τη βιβλιοθήκη της «ΓΛΩΣΣΑΣ», το οποίο ονομάζεται ΓΡΑΨΕ_ΛΕΞΗ(P) με τυπική παράμετρο πίνακα P[8] χαρακτήρων το οποίο θα εμφανίζει στην οθόνη τους χαρακτήρες που περιέχει ο πίνακας. (Το υποπρόγραμμα αυτό δεν θα το κατασκευάσετε, απλώς θα το καλείτε κάθε φορά για να εμφανίζει την άγνωστη λέξη.)

μονάδες 12

Γ4. Το παιχνίδι θα τελειώνει όταν βρεθεί η λέξη, εμφανίζοντας τις προσπάθειες που χρειάστηκε ο χρήστης για να τη βρει, είτε αν δοθούν 8 λανθασμένα γράμματα, οπότε θα αποκαλύπτει την άγνωστη λέξη.

μονάδες 5

Διευκρινίσεις:

- Τα γράμματα που πρέπει να βρεθούν είναι 6.
- Δεν χρειάζεται έλεγχος των δεδομένων που εισάγονται.

- Θεωρείστε ότι ο χρήστης δεν ξαναδίνει το ίδιο γράμμα δεύτερη φορά.
- Η λέξη μπορεί να έχει πάνω από μια φορά το ίδιο γράμμα.
- Αν δοθεί γράμμα που υπάρχει περισσότερες από μία φορές στη λέξη, τότε θεωρείστε ότι έχουν βρεθεί με μια προσπάθεια τόσα γράμματα όσα περιέχει η λέξη.

ΘΕΜΑ Δ

Το Αριθμολαχείο (Λόττο) είναι ένα παιχνίδι τύχης με επιλογή αριθμών. Αφορά στην ακριβή πρόβλεψη έξι (6) διαφορετικών αριθμών που κληρώνονται από μία σειρά σαράντα εννέα (49) αριθμών, από το 1 μέχρι και το 49. Μέχρι σήμερα έχουν γίνει 2300 κληρώσεις.

Να αναπτύξετε αλγόριθμο ο οποίος δίνει τις εξής στατιστικές πληροφορίες:

Δ1. αρχικά διαβάζει τις εξάδες αριθμών που έχουν κληρωθεί μέχρι σήμερα και τις αποθηκεύει σε πίνακα $\Lambda[2300, 6]$

(βλέπε επόμενο σχήμα-παράδειγμα).

Δεν απαιτείται έλεγχος εγκυρότητας των δεδομένων.

Μονάδα 1

| | Κληρωμένες εξάδες | | | | | |
|---------------------------|-------------------|-----|-----|-----|-----|-----|
| 1 ^η κλήρωση | 22 | 43 | 18 | 23 | 16 | 7 |
| 2 ^η κλήρωση | 39 | 11 | 38 | 23 | 24 | 26 |
| ... | ... | ... | ... | ... | ... | ... |
| 2300 ^η κλήρωση | 12 | 10 | 24 | 28 | 38 | 43 |

Δ2. Εμφανίζει τους 6 αριθμούς κάθε κλήρωσης, στην οποία κληρώθηκαν έξι ζυγοί ή έξι μονοί αριθμοί, διαφορετικά αν δεν συνέβη ποτέ αυτό, να εμφανίζει σχετικό μήνυμα.

Μονάδες 3

Δ3. Εμφανίζει τον αριθμό ή τους αριθμούς, αν είναι περισσότεροι του ενός, που έχουν καθυστερήσει περισσότερο να κληρωθούν.

Μονάδες 5

Δ4. Βρίσκει τους 6 αριθμούς που έχουν κληρωθεί τις λιγότερες φορές μέχρι σήμερα. Να τους εμφανίζει με αύξουσα διάταξη ως προς το πλήθος κληρώσεών τους.

Μονάδες 6

Δ5. Να βρίσκει πόσοι το πολύ απ' τους 6 αριθμούς που έχουν κληρωθεί τις περισσότερες φορές μέχρι σήμερα, έχουν κληρωθεί όλοι μαζί σε μια κλήρωση.

Μονάδες 5

Σημείωση: Θεωρείστε ότι και οι 49 αριθμοί έχουν διαφορετικό πλήθος κληρώσεων μεταξύ τους.