

Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον

Θέματα προσομοίωσης

Θέμα Α

A1

Να γράψετε τον αριθμό καθεμιάς από τις παρακάτω προτάσεις και δίπλα τη λέξη Σωστό, αν είναι σωστή, ή τη λέξη Λάθος, αν είναι λανθασμένη.

1. Με τη λειτουργία της συγχώνευσης, δύο ή περισσότερες δομές δεδομένων συνενώνονται σε μία ενιαία δομή.
2. Ο τρόπος κλήσης των διαδικασιών και των συναρτήσεων είναι ίδιος, ενώ ο τρόπος σύνταξής τους είναι διαφορετικός.
3. Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μία έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες. Πχ παρτε την εκφραση $15+2>23-4$
4. Η έννοια του αλγορίθμου συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής.
5. Κάθε βρόχος που υλοποιείται με την εντολή ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ μπορεί να γραφεί και με χρήση της εντολής ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ.
6. Ο δομημένος προγραμματισμός κάνει εκτενή χρήση της GOTO.
7. Ο μεταφραστής (compiler) ανιχνεύει και λογικά και συντακτικά λάθη.
8. Οι τυπικές γλώσσες επεκτείνονται με την ανάπτυξη υποπρογραμμάτων (διαδικασιών και συναρτήσεων).
9. Ένας πίνακας μπορεί να αλλάξει μέγεθος κατά τη διάρκεια εκτέλεσης ενός προγράμματος.
10. Ο ίδιος πίνακας μπορεί να περιέχει ταυτόχρονα και ακεραίους (πχ 12,34,23) και ονόματα (πχ 'Κώστας', 'Γιάννης').

A2.

Για ποιες από τις παρακάτω περιπτώσεις μπορεί να χρησιμοποιηθεί συνάρτηση:

- α. εισαγωγή ενός δεδομένου
- β. υπολογισμός του μικρότερου από πέντε ακεραίους
- γ. υπολογισμός των δύο μικρότερων από πέντε ακεραίους
- δ. έλεγχος αν δύο αριθμοί είναι ίσοι
- ε. ταξινόμηση πέντε αριθμών
- στ. έλεγχος αν ένας χαρακτήρας είναι φωνήεν ή σύμφωνο.

A3.

Απαριθμήστε τις βασικές επεξεργασίες (λειτουργίες) πινάκων.

A4.

Αναφέρετε σύντομα τα πλεονεκτήματα του τμηματικού προγραμματισμού.

Θεμα Β

B1. Γράψτε ακριβώς τι εμφανίζει ο πιο κάτω κώδικας

Αλγόριθμος ΒΧΧΙV

$C[1] \leftarrow 2$

$C[2] \leftarrow 5$

$C[3] \leftarrow 15$

$C[4] \leftarrow -1$

$C[5] \leftarrow 32$

$C[6] \leftarrow 14$

$\min \leftarrow 100$

$\max \leftarrow -100$

Για i από 1 μέχρι 6 με_βήμα 2

$A \leftarrow C[i]$

$B \leftarrow C[i+1]$

Αν $A < B$ τότε

$L_{\min} \leftarrow A$

$L_{\max} \leftarrow B$

αλλιώς

$L_{\min} \leftarrow B$

$L_{\max} \leftarrow A$

Τέλος_αν

Αν $L_{\min} < \min$ τότε

$\min \leftarrow L_{\min}$

Τέλος_αν

Αν $L_{\max} > \max$ τότε

$\max \leftarrow L_{\max}$

Τέλος_αν

Εμφάνισε 'A=', A, ' B=', B, ' Lmin=', Lmin, ' Lmax=', Lmax

Εμφάνισε ' min=', min, ' max=', max

Εμφάνισε '#####'

Τέλος_επανάληψης

$D \leftarrow \max * \min$

Εμφάνισε 'D=', D

Τέλος ΒΧΧΙV

ΦΡΟΝΤΙΣΤΗΡΙΟ ΜΕΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ

Πυξίδα



B2. Μετατρέψτε τον αλγόριθμο/πρόγραμμα αύξουσας ταξινόμησης πίνακα 20 ακεραίων με τη μέθοδο της φουσαλίδας ώστε να σταματά τους ελέγχους όταν πλέον ο πίνακας έχει ταξινομηθεί. Συμβουλή: θα χρειαστεί να αλλάξετε το είδος της μιας από τις 2 επαναλήψεις ΓΙΑ του αλγόριθμου του σχολικού βιβλίου.

Θεμα Γ

Ένα εργοστάσιο κατασκευής playstation παράγει 100 κομμάτια την ημέρα. Κάθε κομμάτι ελέγχεται σε 7 σημεία. Σε πίνακα $\Pi[7, 100]$ καταχωρείται καθημερινά το αποτέλεσμα του ελέγχου, με τους χαρακτήρες 'O' για OK, S για υποπτο βλαβης (suspect) και F για προβληματικό (faulty). Φτιάξτε πρόγραμμα που διαβάζει τον πίνακα αυτό, με έλεγχο εισόδου και κατάλληλα μηνύματα. Στο πρόγραμμα να υλοποιήσετε και συνάρτηση που μετρά πόσες συσκευές από τις 100 έχουν OK σε 6 ή και στα 7 σημεία, επομένως είναι έτοιμες για πώληση. Το πρόγραμμα κάνει καταχώριση στοιχείων και καλεί τη συνάρτηση για 25 μέρες που λειτουργεί το εργοστάσιο (σε διάστημα ενός μήνα) και για κάθε μέρα βγάζει **μήνυμα** πόσες συσκευές είναι έτοιμες προς πώληση.

Συμβουλή. Φτιάξτε πρώτα τη συνάρτηση. Μετά υλοποιήστε την καταχώριση στοιχείων με τον έλεγχο εισόδου και μετά κλείστε σε επανάληψη 25 βημάτων την καταχώριση στοιχείων, την κλήση της συνάρτησης με τον κατάλληλο τρόπο και το **μήνυμα**.

Θέμα Δ.

Γράψτε πρόγραμμα που διαβάζει ακεραίους (άγνωστου πλήθους) μέχρι να διαβαστεί ο αριθμός 999. Οι έγκυροι αριθμοί θα πρέπει είναι στο διάστημα $[1, 500]$ εκτός από τον τελευταίο βέβαια, που θα επιτρέπεται να είναι το 999. Το πρόγραμμα στο τέλος θα εμφανίζει τους 3 μεγαλύτερους σε φθίνουσα σειρά. Θεωρούμε ότι όλοι οι αριθμοί είναι διαφορετικοί μεταξύ τους. Χρησιμοποιήστε μια διαδικασία που θα δέχεται τον κάθε νέο αριθμό που εισάγουμε καθώς και τους ήδη 3 μεγαλύτερους και ανάλογα με την τιμή του νέου αριθμού, θα τον εισάγει ή όχι στην 3άδα των μεγαλύτερων, στο κατάλληλο σημείο (ανάλογα με την τιμή του σε σχέση με τους άλλους 3). Μπορείτε για ευκολία να χρησιμοποιήσετε άλλη μια διαδικασία φθίνουσας ταξινόμησης των 3 μεγαλύτερων ακεραίων.

Προσοχή: επειδή δε γνωρίζουμε από πριν το πλήθος των ακεραίων που θα εισαχθούν, δεν μπορεί να χρησιμοποιηθεί πίνακας για την αποθήκευση των εισαγόμενων αριθμών.

Καλή επιτυχία!