

ΠΛΗ10 – ΤΕΣΤ 36

Θέμα 3: Ερωτήματα σε Γλώσσα C

Ερώτημα 1

Να γραφεί στη γλώσσα προγραμματισμού C η συνάρτηση:

```
int odd_even(int bin[], int n)
```

που δέχεται ως είσοδο τον πίνακα `bin[]`, ο οποίος περιέχει n ακραίους αριθμούς που αντιπροσωπεύουν τα ψηφία ενός δυαδικού αριθμού. Οι ακέραιοι αριθμοί θα πρέπει να έχουν τιμές 0 και 1. Η συνάρτηση επιστρέφει: 1 (αν το πλήθος των 1 στον πίνακα είναι περιττός αριθμός), 2 (αν το πλήθος των 1 στον πίνακα είναι άρτιος αριθμός) ή 0 (αν η είσοδος δεν είναι δυαδικός αριθμός, δηλαδή αν στον πίνακα περιέχονται και άλλες τιμές πλην των 0 και 1).

Ερώτημα 2

Ταξιδεύοντας - ένα παιχνίδι που μπορεί να σας συντροφεύει κατά τη διάρκεια των οδικών σας μετακινήσεων:

Διαβάζετε τα τέσσερα αριθμητικά ψηφία της πινακίδας του προπορευόμενου οχήματος.

Προσθέτετε τα ψηφία αυτά.

Αν το εξαγόμενο της παραπάνω πρόσθεσης είναι μονοψήφιος αριθμός, τότε ανακοινώνετε τον αριθμό αυτό ως τελικό αποτέλεσμα.

Αν το εξαγόμενο της παραπάνω πρόσθεσης είναι διψήφιος αριθμός, τότε βρίσκετε το άθροισμα των δύο ψηφίων του και ανακοινώνετε αυτό το άθροισμα ως τελικό αποτέλεσμα.

Με βάση την παραπάνω περιγραφή του παιχνιδιού, υλοποιήστε πρόγραμμα στη γλώσσα προγραμματισμού C, το οποίο διαβάζει 4 ακέραιους αριθμούς που αντιστοιχούν στα 4 ψηφία της πινακίδας του προπορευόμενου οχήματος (εφαρμόστε αμυντικό προγραμματισμό για να διασφαλίσετε ότι οι αριθμοί που εισάγονται είναι μονοψήφιοι) και στη συνέχεια υπολογίζει το τελικό εξαγόμενο με τον τρόπο που περιγράφεται παραπάνω.

Ερώτημα 3

Να γραφεί στη γλώσσα προγραμματισμού C η συνάρτηση:

```
void compute_average (int pin[N])
```

η οποία να δέχεται ως είσοδο έναν μονοδιάστατο πίνακα ακεραίων pin, με N στοιχεία, και να υπολογίζει και να εκτυπώνει για κάθε θέση του πίνακα τον μέσο όρο των στοιχείων από τη θέση αυτή μέχρι το τέλος του πίνακα. Για κάθε στοιχείο, η εμφάνιση στην οθόνη να έχει τη μορφή:

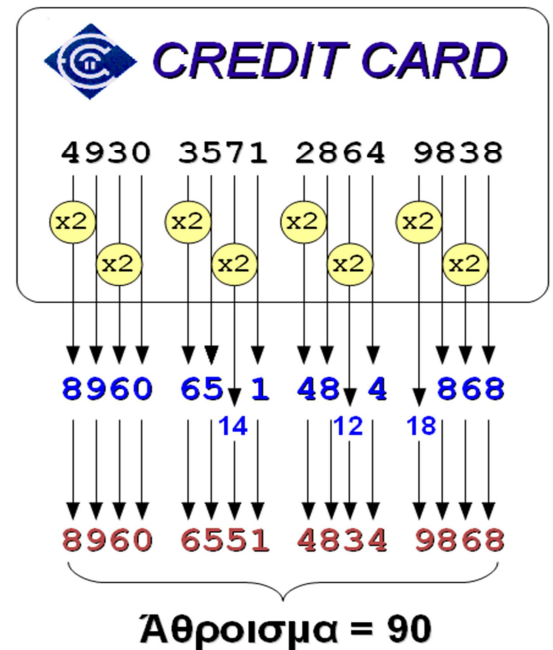
Ο μέσος όρος των στοιχείων από τη θέση k μέχρι το τέλος του πίνακα είναι x

Ερώτημα 4

Να υλοποιηθεί στη γλώσσα προγραμματισμού C συνάρτηση που θα δέχεται ως είσοδο έναν μονοδιάστατο πίνακα 16 ακεραίων, ο οποίος θα περιέχει τα ψηφία του αριθμού μιας πιστωτικής κάρτας. Η συνάρτηση θα ελέγχει την εγκυρότητα της κάρτας σύμφωνα με τον παρακάτω αλγόριθμο:

“Αν το πρώτο ψηφίο είναι μικρότερο από 4 ή μεγαλύτερο από 7, η κάρτα χαρακτηρίζεται ως μη έγκυρη και σταματάμε τον έλεγχο. Διαφορετικά, διπλασιάζουμε τα ψηφία που βρίσκονται στις περιττές θέσεις του αριθμού (1ο, 3ο, 5ο,..., 15ο). Όσα από αυτά γίνουν διψήφια, προσθέτουμε τα ψηφία τους για να γίνουν μονοψήφια. Αν μετά τις αλλαγές το άθροισμα των 16 ψηφίων είναι ακέραιο πολλαπλάσιο του 10, η κάρτα είναι έγκυρη. Αλλιώς είναι μη έγκυρη.”

Η συνάρτηση που θα φτιάξετε θα πρέπει να επιστρέφει 1 αν ο αριθμός είναι έγκυρος και 0 αν είναι μη έγκυρος. Για να ελέγξετε τυχόν σφάλματα στον πίνακα εισόδου (πχ. αν κάποιο στοιχείο είναι αρνητικό ή μεγαλύτερο από 9), χρησιμοποιήστε αμυντικό προγραμματισμό. Σε τέτοιες περιπτώσεις η συνάρτηση θα επιστρέφει -1. Στο σχήμα παραπλεύρως φαίνεται η διαδικασία που περιγράφηκε ανωτέρω (δεν φαίνεται ο αρχικός έλεγχος αν το πρώτο ψηφίο είναι μεταξύ 4 και 7, ούτε ο τελικός έλεγχος αν το άθροισμα που προκύπτει είναι πολλαπλάσιο του 10). Είναι εμφανές ότι η κάρτα του σχήματος είναι έγκυρη.



Θέμα 4: Ερωτήματα σε Δομές Δεδομένων

Ερώτημα 1

Έστω δυαδικό δέντρο αναζήτησης με προδιατεταγμένη διαπέραση: 5, 2, 1, 4, 3, 6, 10, 8, 9.

α) Σχεδιάστε το δέντρο.

β) Σχεδιάστε τη συνεχόμενη αναπαράστασή του.

γ) Πόσοι έλεγχοι θα γίνουν για να διαπιστωθεί ότι δεν υπάρχει ο αριθμός 7 στο δέντρο; Δικαιολογήστε την απάντησή σας.

δ) Γράψτε στη γλώσσα προγραμματισμού C τη συνάρτηση:

```
void print_ancestors(int A[], int t)
```

η οποία δέχεται ως είσοδο έναν μονοδιάστατο πίνακα ακεραίων A που εκφράζει τη συνεχόμενη αναπαράσταση ενός δυαδικού δέντρου και έναν ακέραιο t που αντιστοιχεί στη θέση ενός κόμβου μέσα στον A. Η συνάρτηση θα πρέπει να τυπώνει όλους τους προγόνους αυτού του κόμβου (θέση και τιμή) μέχρι και τη ρίζα του δέντρου. Δεν απαιτείται να κάνετε προληπτικούς ελέγχους για τις τιμές του πίνακα και για την τιμή του t. Θεωρήστε ότι οι τιμές των κόμβων του δέντρου θα είναι πάντα μη αρνητικές, όμως ο πίνακας A θα έχει τιμές -1 στις κενές θέσεις (όπου δεν υπάρχει κόμβος). Το t θα είναι πάντα μεταξύ 0 και $\text{len}(A) - 1$ (όπου $\text{len}(A)$ δηλώνει το μέγεθος του πίνακα A) και θα αντιστοιχεί πάντα σε υπάρχοντα κόμβο.

Ερώτημα 2

Να γράψετε στη γλώσσα προγραμματισμού C τη συνάρτηση:

```
int count(struct node *list)
```

η οποία δέχεται ως παράμετρο μια απλά συνδεδεμένη λίστα ακεραίων αριθμών και επιστρέφει το πλήθος των στοιχείων της συνδεδεμένης λίστας. Αν η λίστα είναι κενή η συνάρτηση θα επιστρέφει -1.

Η δομή των κόμβων της λίστας δίνεται ως:

```
struct node {  
    int key;  
    struct node *next;  
};
```

Ερώτημα 3

Να γραφεί στη γλώσσα προγραμματισμού C η συνάρτηση:

```
int winner(int votes[], int n)
```

που δέχεται ως είσοδο τον πίνακα votes[], ο οποίος περιέχει n ακραίους αριθμούς. Οι ακέραιοι αριθμοί θα πρέπει να έχουν τιμές 1 και 2. Η συνάρτηση επιστρέφει:

1, αν στον πίνακα επικρατεί η τιμή 1 (δηλαδή τα 1 είναι περισσότερα από τα 2)

2, αν στον πίνακα επικρατεί η τιμή 2 (δηλαδή τα 2 είναι περισσότερα από τα 1)

3, αν υπάρχει ισοψηφία στον πίνακα (δηλαδή τα 1 είναι ίσα με τα 2)

-1, αν στον πίνακα περιέχονται και άλλες τιμές πλην των 1 και 2