

ΠΛΗ10 – ΤΕΣΤ 35

Θέμα 1: Εισαγωγή στους Η/Υ

Ερώτημα 1

Μετατρέψτε τον αριθμό $(1A3.2)_{16}$ από το δεκαεξαδικό στο οκταδικό σύστημα με δύο τρόπους

- i) μέσω του δυαδικού,
- ii) μέσω του δεκαδικού συστήματος.

Θέμα 2: Αλγόριθμοι και Ψευδογλώσσα

Ερώτημα 1

Σχεδιάστε το Διάγραμμα Ροής Προγράμματος (ΔΡΠ) αλγορίθμου, που θα δέχεται ως είσοδο δύο μονοδιάστατους πίνακες ακεραίων και θα τυπώνει τα κοινά στοιχεία τους. Οι πίνακες εισόδου θα είναι πάντα ταξινομημένοι κατά αύξουσα σειρά και δεν θα περιέχουν διπλά στοιχεία. Δεν απαιτείται να εφαρμόσετε αμυντικό προγραμματισμό, όμως χρησιμοποιήστε αυτή την πληροφορία για να ελαχιστοποιήσετε το πλήθος των ελέγχων του αλγορίθμου σας..

Ερώτημα 2

Θεωρήστε τον παρακάτω αλγόριθμο σε ψευδοκώδικα που χρησιμοποιεί μια ουρά FIFO.

- Η διαδικασία QUEUE-INIT() αρχικοποιεί την ουρά.
- Η διαδικασία QUEUE-PUT() εισάγει ένα στοιχείο στην ουρά.
- Η συνάρτηση QUEUE-GET() διαγράφει ένα στοιχείο από την ουρά και επιστρέφει την τιμή του.
- Η συνάρτηση QUEUE-IS-EMPTY() επιστρέφει 0 αν η ουρά είναι άδεια.

ΠΡΟΓΡΑΜΜΑ FIFO

ΔΕΔΟΜΕΝΑ

I, X: INTEGER;

ΑΡΧΗ

ΥΠΟΛΟΓΙΣΕ QUEUE-INIT();

ΓΙΑ I:=0 ΕΩΣ 9 ΕΠΑΝΑΛΑΒΕ

ΥΠΟΛΟΓΙΣΕ QUEUE-PUT(I);

ΓΙΑ ΤΕΛΟΣ

ΕΝΟΣΩ (QUEUE-IS-EMPTY() <> 0)

ΓΙΑ I:=0 ΕΩΣ 2 ΕΠΑΝΑΛΑΒΕ

X:=QUEUE-GET();

ΥΠΟΛΟΓΙΣΕ QUEUE-PUT(X);

ΓΙΑ ΤΕΛΟΣ

ΤΥΠΩΣΕ QUEUE-GET();

ΕΝΟΣΩ ΤΕΛΟΣ

ΤΕΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Τι τυπώνει το συγκεκριμένο πρόγραμμα κατά την εκτέλεσή του;

Θέμα 3: Ερωτήματα σε Γλώσσα C

Ερώτημα 1

Γράψτε στη γλώσσα προγραμματισμού C τη συνάρτηση:

```
void l_and(int a[], int b[], int n)
```

που δέχεται ως είσοδο τους πίνακες $a[]$ και $b[]$, ο καθένας απ' τους οποίους περιέχει n ακέραιους αριθμούς με τιμές 0 και 1 που αντιπροσωπεύουν τα ψηφία ενός δυαδικού αριθμού, και εμφανίζει στην οθόνη τα ψηφία της λογικής πράξης AND των δύο δυαδικών αριθμών.

.

Ερώτημα 2

Γράψτε στη γλώσσα προγραμματισμού C τη συνάρτηση:

```
void dec2hex(int dec, char hex[], int N)
```

που δέχεται ως είσοδο τον ακέραιο αριθμό `dec` του δεκαδικού συστήματος και τον μετατρέπει στο δεκαεξαδικό σύστημα. Τα ψηφία του δεκαεξαδικού αριθμού εισάγονται στον πίνακα χαρακτήρων `hex`, `N` στοιχείων, και στη συνέχεια εκτυπώνεται ο δεκαεξαδικός αριθμός. Θεωρήστε ότι ο δεκαεξαδικός αριθμός δεν έχει περισσότερα από `N` ψηφία

Ερώτημα 3

Να υλοποιηθεί αναδρομική συνάρτηση στη γλώσσα προγραμματισμού C, η οποία δέχεται ως ορίσματα δύο ακέραιους αριθμούς, m και n ($n \geq 0$), και επιστρέφει τη νιοστή δύναμη του m , δηλαδή το m^n .

Ερώτημα 4

Δίνεται η ακόλουθη αναδρομική συνάρτηση:

```
int xyz(int a, int b)
{
    if (a == b)
        return(0);
    else
        if (a > b)
            return (xyz(a - 1, b) + 1);
        else
            return (xyz(a, b - 1) + 1);
}
```

i) Συμπληρώστε τον ακόλουθο πίνακα με τις τιμές που επιστρέφει η συνάρτηση για τις δοσμένες τιμές των παραμέτρων της:

a	b	Τιμή που επιστρέφεται
3	3	
6	5	
5	6	
8	12	
12	8	

ii) Περιγράψτε, με λίγα λόγια, δοθέντων των δύο ακεραίων a και b, τι υπολογίζει η συνάρτηση αυτή.

Θέμα 4: Ερωτήματα σε Δομές Δεδομένων

Ερώτημα 1

Έστω ένα πλήρες δυαδικό δέντρο με συνεχόμενη αναπαράσταση: 10, 9, 7, 5, 2, 6, 8, 3, 4, 1

- α)** Εξηγήστε γιατί το δέντρο δεν είναι δέντρο-σωρός. Στη συνέχεια αντιμετωπίστε τα περιεχόμενα ορισμένων κόμβων του δέντρου ώστε να γίνει δέντρο-σωρός, με όσο το δυνατόν λιγότερες αντιμεταθέσεις. Σχεδιάστε το παραγόμενο δέντρο-σωρό.
- β)** Από το δέντρο-σωρό που προέκυψε στο ερώτημα (α) διαγράψτε τον κόμβο της ρίζας, δίνοντας τη διαδικασία διαγραφής βήμα προς βήμα.
- γ)** Υλοποιήστε συνάρτηση στη γλώσσα προγραμματισμού C, η οποία θα δέχεται ως είσοδο έναν μονοδιάστατο πίνακα με τη συνεχόμενη αναπαράσταση ενός δέντρου-σωρού, το πλήθος των κόμβων του δέντρου-σωρού και ένα νέο στοιχείο x και θα εισάγει το x στο δέντρο-σωρό, ενημερώνοντας κατάλληλα τον πίνακα που αναπαριστά το δέντρο-σωρό. Θεωρήστε ότι ο πίνακας δεν είναι πλήρης και συνεπώς υπάρχει χώρος για την προσθήκη του νέου στοιχείου.

Ερώτημα 2

Υποθέστε ότι μια συνδεδεμένη λίστα αποτελείται από κόμβους του τύπου

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

Γράψτε τη συνάρτηση

```
void average(struct node *list)
```

που δέχεται ως είσοδο τον δείκτη list που δείχνει στον πρώτο κόμβο μιας συνδεδεμένης λίστας και τυπώνει τον μέσο όρο των τιμών key των στοιχείων που βρίσκονται στις περιττές θέσεις της λίστας (δηλαδή στην 1^η, στην 3^η, στην 5^η κλπ. θέσεις της λίστας).