

ΠΛΗ10 – ΤΕΣΤ 27

Θέμα 2: Ψευδογλώσσα και Αλγόριθμοι

Ερώτημα 1

Δώστε το Διάγραμμα Ροής Προγράμματος για τον υπολογισμό του κάτωθι αθροίσματος:

$S = 1! - 2! + 3! - 4! + \dots - 100!$, όπου $N! = 1*2*3*\dots*N$

Θέμα 3: Ερωτήματα σε Γλώσσα C

Ερώτημα 1

Γράψτε στη γλώσσα προγραμματισμού C τη συνάρτηση:

```
void bitwise_XOR(int a[], int b[], int N)
```

που δέχεται ως είσοδο τους πίνακες $a[]$ και $b[]$, ο καθένας από τους οποίους περιέχει N ακέραιους αριθμούς με τιμές 0 και 1 που αντιπροσωπεύουν τα ψηφία ενός δυαδικού αριθμού, και εμφανίζει στην οθόνη το αποτέλεσμα της λογικής πράξης XOR μεταξύ των αντίστοιχων ψηφίων των δύο δυαδικών αριθμών, ξεκινώντας από τα ψηφία στην αρχή του κάθε πίνακα.

Ερώτημα 2

Να γραφεί πρόγραμμα στη γλώσσα προγραμματισμού C που διαβάζει τα στοιχεία ενός πίνακα ακεραίων αριθμών `int a[2*N]` και αντιγράφει το πρώτο μισό του πίνακα `a[]` (στοιχεία από τις θέσεις 0 έως και $N-1$) στον πίνακα `int b[N]` με την ίδια σειρά (δηλαδή, το πρώτο στοιχείο του πίνακα `a[]` στην πρώτη θέση του πίνακα `b[]` κ.ο.κ.) και το δεύτερο μισό του πίνακα `a[]` (στοιχεία από τις θέσεις N έως και $2*N-1$) στον πίνακα `int c[N]`, αλλά με αντίστροφη σειρά (δηλαδή, το τελευταίο στοιχείο του `a[]` στην πρώτη θέση του `c[]` κ.ο.κ.).

Ερώτημα 3

Τι τυπώνει το ακόλουθο πρόγραμμα; Αιτιολογήστε την απάντησή σας.

```
#include <stdio.h>
void main()
{
    int table[]={1, 2, 4, 6}, *p, *q;
    p = table;
    q = p+2;
    printf("%d %d %d %d\n", table[1], *p, *(q+1), *q+1);
}
```

Ερώτημα 4

Γράψτε μια συνάρτηση στη γλώσσα προγραμματισμού C που θα δέχεται ως ορίσματα 3 πραγματικούς αριθμούς A, B, C και θα επιστρέφει το εμβαδόν του τριγώνου με μήκη πλευρών A, B, C.

Ο υπολογισμός του εμβαδού θα γίνεται με τον τύπο του Ήρωνα:

$$E = \sqrt{T(T - A)(T - B)(T - C)}$$

όπου

$$T = \frac{A + B + C}{2}$$

Η συνάρτηση θα πρέπει να ελέγχει τα εξής:

- Αν τα A, B, C είναι θετικοί αριθμοί (εάν δεν είναι, η συνάρτηση θα επιστρέφει -1).
- Αν τα A, B, C αποτελούν μήκη πλευρών τριγώνου. Ο έλεγχος γίνεται με την τριγωνική ανισότητα: το μήκος κάθε πλευράς πρέπει να είναι μικρότερο από (ή ίσο με) το άθροισμα των μηκών των άλλων δύο πλευρών (σημειώστε ότι σε περίπτωση ισότητας το τρίγωνο έχει εμβαδόν 0, αλλά συνεχίζει να θεωρείται τρίγωνο). Σε περίπτωση που τα A, B, C είναι θετικά αλλά δεν μπορούν να αποτελούν πλευρές τριγώνου, η συνάρτηση να επιστρέφει -2.

Υπόδειξη: Για τον υπολογισμό της τετραγωνικής ρίζας μπορείτε να χρησιμοποιήσετε τη συνάρτηση βιβλιοθήκης sqrt().

Θέμα 4: Ερωτήματα σε Δομές Δεδομένων

Ερώτημα 1

Δίνονται κατά σειρά οι ακέραιες τιμές: 4, 5, 7, 1, 6, 2, 8, 3.

- i. Σχηματίστε το πλήρες δυαδικό δένδρο αναζήτησης, οι κόμβοι του οποίου έχουν τις τιμές αυτές.
- ii. Δώστε την προ-διατεταγμένη και τη μετα-διατεταγμένη διαπέραση του δέντρου.

Ερώτημα 2

Για ένα Δυαδικό Δέντρο, η προ-διατεταγμένη και η ενδο-διατεταγμένη διαπέραση έχουν τα εξής αποτελέσματα:

ΠΡΟ-ΔΙΑΤΕΤΑΓΜΕΝΗ: 5, 3, 2, 6, 1, 4, 7

ΕΝΔΟ-ΔΙΑΤΕΤΑΓΜΕΝΗ: 3, 5, 4, 1, 6, 2, 7

Σχεδιάστε το Δυαδικό Δέντρο.

Ερώτημα 3

Δίνεται η παρακάτω δήλωση για κόμβους δυαδικού δένδρου:

```
struct treenode {  
    int data;  
    struct treenode *left;  
    struct treenode *right;  
};
```

όπου το left δείχνει στο αριστερό παιδί και το right δείχνει στο δεξί παιδί του κόμβου.

Να γραφεί συνάρτηση `struct treenode *split_left_subtree(struct treenode *t)` στη γλώσσα προγραμματισμού C που αποκόπτει το αριστερό υποδένδρο της ρίζας του δυαδικού δένδρου `t` και επιστρέφει δείκτη στη ρίζα του υποδένδρου που έχει αποκοπεί. Η ρίζα του δένδρου `t` μετά την ολοκλήρωση της συνάρτησης δεν πρέπει να έχει αριστερό παιδί (δηλαδή το αριστερό παιδί να είναι NULL). Θεωρήστε πως η παράμετρος `t` δεν είναι ποτέ NULL και έτσι δεν χρειάζεται σχετικός έλεγχος για αμυντικό προγραμματισμό.