

**Θέμα 1<sup>ο</sup> - Πολυπλοκότητα (12 Μονάδες)**

Υποθέτοντας ότι κάθε βασική λειτουργία ( $t_{\text{fetch}}$ ,  $t_{\text{store}}$ ,  $t_C$ ,  $t_H$ , κ.λ.π.) εκτελείται σε μία (1) μονάδα χρόνου, υπολογίστε το χρόνο εκτέλεσης του παρακάτω κώδικα. Αφού υπολογίσετε το χρόνο εκτέλεσης ως συνάρτηση της μεταβλητής  $n$ , σχολιάστε ποιο είναι το (σφικτό) άνω ασυμπτωτικό όριο  $O()$  του χρόνου εκτέλεσης αυτού του κώδικα.

```
int K=0;

for (int i=0; i<n; ++i)

    for (int j=1; j<n; ++j)

        ++K
```

Σημ.  $S_n = \sum_{i=1}^n i = \frac{n(n+1)}{2}$

**Θέμα 2<sup>ο</sup> – Στοιβά (10 Μονάδες)**

Έστω ο αφηρημένος τύπος δεδομένων `stack` για την παράσταση μίας στοίβας ακέραιων αριθμών, ο οποίος υποστηρίζει τις ακόλουθες πράξεις:

<code>int isEmpty</code>	<code>(stack s);</code>	<code>/*Ελέγχει αν η στοίβα είναι άδεια*/</code>
<code>void push</code>	<code>(stack *s, int x);</code>	<code>/*Εισάγει ένα στοιχείο στη στοίβα*/</code>
<code>int pop</code>	<code>(stack *s);</code>	<code>/*Αφαιρεί ένα στοιχείο από τη στοίβα*/</code>

Συμπληρώστε το σώμα της συνάρτησης `product` η οποία δέχεται μία στοίβα ως παράμετρο και την αδειάζει, βρίσκοντας συγχρόνως το γινόμενο των στοιχείων.

```
int product (stack *s)

{

}
```

**Θέμα 3<sup>ο</sup> – Κλήση Αναδρομικής Συνάρτησης (8 Μονάδες)**

```
void f (int n)

{

    printf("%d", n);
    if (n==1)
        printf("ΤΕΛΟΣ\n");
    else if (n%2 ==0);
        f(n/2);
    else
        f(3*n+1);
    /*β ερώτηση*/
}
```

A) Τι θα τυπωθεί αν κληθεί η συνάρτηση αυτή με παράμετρο την τιμή 10

B) Τι θα τυπωθεί αν μεταφέρουμε το `printf("%d", n);` μετά το σχόλιο `/*β ερώτηση*/`

#### Θέμα 4<sup>ο</sup> – Συνδεδεμένη Λίστα (20 Μονάδες)

Δίνεται μία απλά συνδεδεμένη λίστα list που περιέχει ως πληροφορία λέξεις. . κάθε λέξη αποτελείται από 19 ή λιγότερα πεζά λατινικά γράμματα. Η λίστα αυτή είναι ταξινομημένη σε λεξικογραφική διάταξη, αλλά πιθανώς να περιέχει κάποιες λέξεις περισσότερες από μία φορά. Ζητείται να γράψετε μία συνάρτηση C η οποία να δέχεται ως παράμετρο αυτή τη λίστα και να αφαιρεί από αυτήν τα στοιχεία που εμφανίζονται περισσότερες φορές. Σε περίπτωση όμως που αφαιρεθεί κάποια λέξη, θα πρέπει να αποδεσμεύεται ο χώρος στη μνήμη.

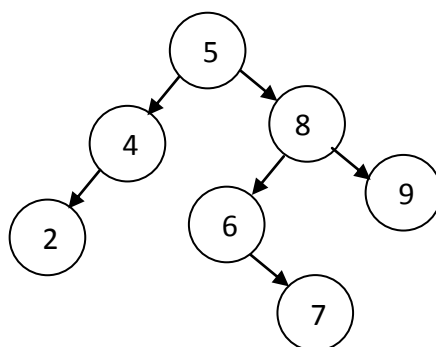
#### Θέμα 5<sup>ο</sup> – Δυαδικό Δένδρο (15 Μονάδες)

A (12 Μονάδες)

Δίνεται ένα δυαδικό δένδρο t. Ζητείται να γραφεί μία συνάρτηση που να παίρνει ως παράμετρο τη δομή του δυαδικού δένδρου t και να υπολογίζει το πλήθος των κόμβων που έχουν ακριβώς δύο παιδιά. Για παράδειγμα αν η συνάρτηση εφαρμοστεί στο δυαδικό δένδρο του σχήματος, θα πρέπει να επιστρέψει τον αριθμό 2

B (3 Μονάδες)

Δώστε μία ακολουθία αριθμών η οποία εάν εισαχθεί σε ένα άδειο δένδρο θα δημιουργήσει το δυαδικό δένδρο αναζήτησης (binary search tree) του σχήματος



#### Θέμα 6<sup>ο</sup> – Ταξινόμηση (10 Μονάδες)

Ας υποθέσουμε ότι έχουμε την ακολουθία ακέραιων αριθμών [2,1,4,1,6,8,3,5].

Χρησιμοποιήστε τον αλγόριθμο ταξινόμησης φυσαλίδας (Bubble Sort) για να ταξινομήσετε την ακολουθία σε αύξουσα σειρά. Καταγράψτε κάθε βήμα του αλγόριθμου δείχνοντας τις ενδιαμέσες μορφές της ακολουθίας καθώς αυτή ταξινομείται κατά τη διάρκεια εφαρμογής του αλγορίθμου. Δώστε μία σύντομη επεξήγηση του τι συμβαίνει στο κάθε βήμα. Αρχίζουμε από την απεικόνιση της ακολουθίας:

2	1	4	1	6	8	3	5
---	---	---	---	---	---	---	---

#### Θέμα 7<sup>ο</sup> – AVL Δένδρα (15 Μονάδες)

Ας θεωρήσουμε την ακολουθία των ακέραιων αριθμών [1,4,6,3,7,2,10]. Σχεδιάστε πως εισάγετε κάθε στοιχείο της ακολουθίας αυτής σε ένα δένδρο AVL, και πως το αναδιατάσσετε κάθε φορά που αυτό χρειάζεται. Τα στοιχεία εισάγονται με τη σειρά που δίνονται στην ακολουθία. Σημειώστε τι είδους αναδιάταξη εφαρμόζετε (ΑΑ, ΔΔ, ΑΔ, ΔΑ)