ΟΝΟΜΑΤΕΠΩΝΥΜΟ: .....
SDI (sdiYYOONNN): .....
GitHub username: .....

Εισαγωγή στον Προγραμματισμό - Σεπτέμβριος 2026

Διάρκεια: 120 λεπτά / Σύνολο: 100 Μονάδες

Τα προγράμματα C που θα γράψετε πρέπει να είναι δομημένα, διατυπωμένα ευκρινώς εντός του διαθέσιμου χώρου και με επαρκή τεκμηρίωση ώστε να είναι κατανοητά.

1. Mystery [10 Μονάδες]

Η συνάρτηση `mystery` δέχεται ως όρισμα έναν ακέραιο αριθμό που αποτελείται από τα 3 τελευταία ψηφία του sdi σας. Για παράδειγμα, αν ο sdi σας είναι ο sdi2500789 τότε καλούμε την συνάρτηση ως `mystery(789)`. Τι θα τυπώσει η συνάρτηση για τα ψηφία του δικού σας sdi; Αιτιολογήστε όπου νομίζετε πως χρειάζεται.

```
int mystery(int number) {
  int i, total = 0;
  number += 100;
  printf("%s: %d, (%d)\n", "Computing total", total, i);
  for(i = 0 ; number ; i++) {
    total = 10 * total + number % 10;
    number /= 10;
    printf("%d: total: %d\n", i, total);
  }
  printf("%d: total: %d number: %d\n", i, total, number);
  return total;
}
```

Απάντηση:

2. Η συνάρτηση cons (15 Μονάδες)

Ζητήσαμε από την Κλαυδία να προγραμματίσει κάτι στα γρήγορα και χρησιμοποίησε την συνάρτηση `cons` στον κώδικά της:

```
int cons(int a[], int n) {
    int best = 1;
    int current = 1;

    for (int i = 1; i <= n; i++) {
        if (a[i] == a[i - 1])
            current++;
        else
            current = 1;

        if (current > best)
            best = current;
    }

    return best;
}
```

Τι κάνει η συνάρτηση cons (μέχρι 15 λέξεις εξήγηση);

Τι θα τυπώσει η παρακάτω ακολουθία εντολών;

```
int arg[] = {4, 4, 2, 6, 7, 7, 7, 3};
printf("%d", cons(arg, 6));
```

Υπάρχει κάποιο σφάλμα στην cons, αν n είναι το μέγεθος του πίνακα;

3. Το μεγαλύτερο άλμα - polevault [25 Μονάδες]

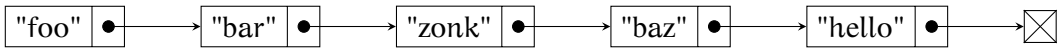
Μας κάλεσαν από το Diamond League να γράψουμε ένα πρόγραμμα υπολογισμού της καλύτερης επίδοσης σε άλματα επί κοντώ. Το πρόγραμμα θα παίρνει τις επιδόσεις των αθλητών/τριών με την σειρά με τις οποίες τις πέτυχαν ως ορίσματα με τον εξής τρόπο: το όνομα του/της κάθε αθλητή/τριας θα ακολουθείται από την επίδοση, ενώ στην πρότυπη έξοδο περιμένουμε να δούμε το άτομο με την καλύτερη επίδοση. Σε περίπτωση ισοβαθμίας, το άτομο που πέτυχε την υψηλότερη επίδοση πρώτο παίρνει το χρυσό. Παράδειγμα εκτέλεσης ακολουθεί:

```
$ ./polevault Kendricks 5.72 Guttormsen 5.82 Karalis 6.12 Marschall 5.82
Out of 4 athletes, the gold goes to Karalis with a jump to 6.12!
```

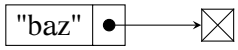
Απάντηση:

4. Ν-οστό Στοιχείο Λίστας [25 Μονάδες]

Γράψτε μια συνάρτηση `nth` η οποία παίρνει ως όρισμα μια λίστα από πίνακες χαρακτήρων τύπου `List` μαζί με έναν αριθμό `n` και επιστρέφει μια νέα λίστα με μοναδικό στοιχείο το n-οστό στοιχείο της λίστας. Αν δεν υπάρχει n-οστό στοιχείο, το πρόγραμμα θέλουμε να τερματίζει με κωδικό εξόδου 1. Για παράδειγμα, αν δοθεί η ακόλουθη λίστα:



και n=3 περιμένουμε να επιστραφεί η ακόλουθη νέα λίστα (0-indexing όπως στην C):



Ποια είναι η χρονική και η χωρική πολυπλοκότητα του αλγορίθμου σας (8/25 της βαθμολογίας); Ο τύπος `List` δίνεται παρακάτω:

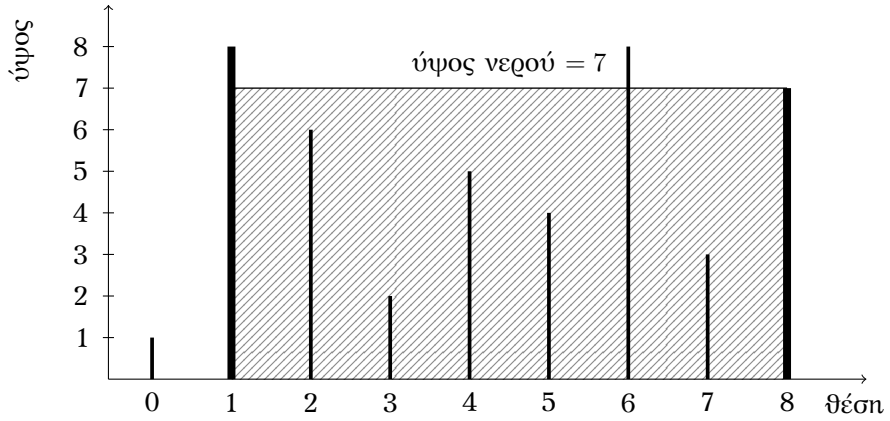
```
typedef struct node {
  char value[32];
  struct node * next;
} * List;
```

Απάντηση:

5. Η Μεγαλύτερη Χωρητικότητα - capacity [25 Μονάδες]

Οι πολιτικοί μηχανικοί του ΕΜΠ χρειάζονται τη βοήθειά μας! Θέλουν να κατασκευάσουν μια σειρά από φράγματα και, προτού ξεκινήσουν τις εργασίες, πρέπει να υπολογίσουν την *μέγιστη* δυνατή ποσότητα νερού που μπορεί να αποθηκευτεί ανάμεσα στις διάφορες βουνοκορφές. Για λόγους απλοποίησης, θεωρούμε ότι έχουμε N κορυφές τοποθετημένες σε ίσες αποστάσεις μεταξύ τους. Η κάθε κορυφή i βρίσκεται στη θέση i και έχει ύψος h_i. Επιλέγοντας δύο κορυφές i και j, με $i < j$, σχηματίζεται μαζί με το έδαφος μια ορθογώνια δεξαμενή, όπως φαίνεται στο Σχήμα 1 (δεν μας πειράζει αν παρεμβάλλονται άλλες βουνοκορφές ενδιάμεσα).

Η στάθμη του νερού δεν μπορεί φυσικά να ξεπεράσει την χαμηλότερη από τις κορυφές που σχηματίζουν την ορθογώνια δεξαμενή. Για τις ανάγκες του προβλήματος θεωρούμε τη χωρητικότητα ως το εμβαδόν του ορθογωνίου που σχηματίζεται από την δεξαμενή.



Σχήμα 1: Παράδειγμα για μια σειρά από κορυφές: $h = [1, 8, 6, 2, 5, 4, 8, 3, 7]$. Οι κορυφές στις θέσεις 1 και 8 σχηματίζουν τη δεξαμενή μέγιστης χωρητικότητας. Η απόστασή τους είναι $8 - 1 = 7$ και το ύψος του νερού είναι 7, επομένως η χωρητικότητα είναι $7 \cdot 7 = 49$.

Γράψτε ένα πρόγραμμα το οποίο παίρνει από την πρότυπη είσοδο στην πρώτη γραμμή το πλήθος N των κορυφών και στην δεύτερη γραμμή τα N ακέραια ύψη των κορυφών $h_0, h_1, \ldots, h_{N-1}$ και υπολογίζει τη **μέγιστη δυνατή χωρητικότητα** που μπορεί να σχηματιστεί από μία δεξαμενή καθώς και τις κορυφές ανάμεσα στις οποίες θα εμφανιστεί αυτή η χωρητικότητα. Ο αλγόριθμός σας θέλουμε να είναι σωστός αλλά και αποδοτικός.

Ποια είναι η χρονική και η χωρική πολυπλοκότητα της λύσης σας; Αιτιολογήστε σύντομα την απάντησή σας (8/25 της βαθμολογίας).

Παράδειγμα εκτέλεσης ακολουθεί:

```
$ cat mountains.txt
9
1 8 6 2 5 4 8 3 7

$ ./capacity < mountains.txt
Maximum Capacity between mountains at positions 1 and 8: (8-1) * 7 = 49
```

Απάντηση:

Απάντηση: