ΟΝΟΜΑΤΕΠΩΝΥΜΟ: .....
SDI (sdiYYOONNN): .....

Εισαγωγή στον Προγραμματισμό - Σεπτέμβριος 2024

Επαναληπτική Εξεταστική

Διάρκεια: 135 λεπτά / Σύνολο: 100 Μονάδες (+5 bonus)

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. About [5 Μονάδες]

Γράψτε μια συνάρτηση `about` η οποία τυπώνει 3 γραμμές στην πρότυπη έξοδο (stdout): 1) το όνομά σας με λατινικούς χαρακτήρες στην 1η, 2) το sdi σας στην 2η και 3) `I'm 100% "ready"!` στην 3η. Παράδειγμα εξόδου από την εκτέλεση της συνάρτησης `about`:

```
Michael Jordan
sdi2300999
I'm 100% "ready"!
```

Απάντηση:

2. Η συνάρτηση what (10 Μονάδες)

```
int what(int *array, int x, int y) {
  int mid, low = 0, high = y - 1;
  while (low <= high) {
    mid = low + (high - low) / 2;
    printf("%d\n", mid);
    if (array[mid] == x) return 1;
    else if (array[mid] < x) low = mid + 1;
    else high = mid - 1;
  }
  return 0;
}
```

Τι κάνει η συνάρτηση what (μέχρι 10 λέξεις εξήγηση); Τι θα τυπώσει αν κληθεί με ορίσματα `{57, 98, 105, 241}`, `36`, `4`;
Απάντηση:

3. Εύρεση Μηδενός σε Πίνακα [15 Μονάδες]

Γράψτε μία συνάρτηση `find_zero` η οποία λαμβάνει ως όρισμα έναν πίνακα από τιμές double ταξινομημένες σε αύξουσα σειρά και επιστρέφει την θέση (index) όπου υπάρχει το 0 ή -1 αν δεν το βρει. Η συνάρτηση μπορεί να έχει οποιαδήποτε διεπαφή (ορίσματα / τύπο επιστροφής) επιθυμείτε. Για παράδειγμα αν δοθεί ο πίνακας `{-8.1, -2.3, 0.0, 1.9}` θέλουμε να επιστραφεί η τιμή 2. Αντίστοιχα, αν δοθεί ο πίνακας `{1.0}` θέλουμε να επιστραφεί -1. Τι χρονική και χωρική πολυπλοκότητα έχει ο αλγόριθμός σας (5/15 της βαθμολογίας);
Απάντηση:

4. Μετρητής Λέξεων [25 Μονάδες]

Γράψτε ένα πρόγραμμα `wordcount` το οποίο διαβάζει ένα κείμενο από την πρότυπη είσοδο (stdin) και τυπώνει στην πρότυπη έξοδο (stdout) τον αριθμό των λέξεων που διάβασε καθώς και το μέσο μήκος λέξης του κειμένου με ακρίβεια δύο δεκαδικών ψηφίων. Ως *λέξη*, ορίζουμε οποιαδήποτε ακολουθία συνεχόμενων χαρακτήρων A-Z, a-z και 0-9 (μετράμε μόνο λέξεις με λατινικούς χαρακτήρες) και ως *μήκος* τον αριθμό των συνεχόμενων χαρακτήρων. Για παράδειγμα, το κείμενο "I'm a good person" έχει σύνολο 5 λέξεις και μέσο μήκος (1 + 1 + 1 + 4 + 6)/5 = 2.6 χαρακτήρες. Παραδείγματα εκτέλεσης ακολουθούν:

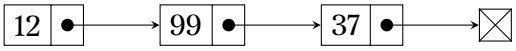
```
$ echo "I'm a good person" | ./wordcount
Total words: 5
Average word length: 2.60
```

```
$ cat quote.txt
"Success is not final, failure is not fatal: It is the courage to continue that counts."
$ ./wordcount < quote.txt
Total words: 16
Average word length: 4.25
```

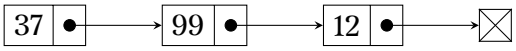
Απάντηση:

5. Αντιστροφή Λίστας [25 Μονάδες]

Γράψτε μια συνάρτηση `reverse_list` η οποία παίρνει ως όρισμα μια λίστα ακεραίων τύπου `List` και επιστρέφει μια νέα λίστα με τους κόμβους της σε αντίστροφη σειρά σε σχέση με την αρχική. Για παράδειγμα, αν δοθεί η ακόλουθη λίστα:



περιμένουμε να επιστραφεί η ανεστραμμένη χωρίς παρενέργειες (side-effects) στην αρχική:



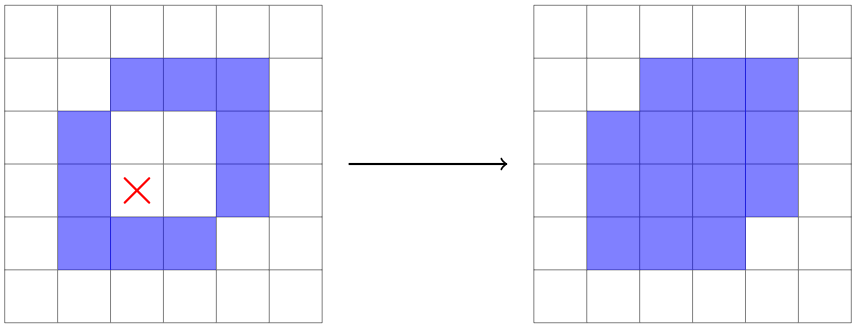
Ποια είναι η χρονική και η χωρική πολυπλοκότητα του αλγορίθμου σας (8/25 της βαθμολογίας); Ο τύπος `List` δίνεται παρακάτω:

```
typedef struct node {
  int value;
  struct node * next;
} * List;
```

Απάντηση:

6. Γεμίζοντας με Χρώμα [25 Μονάδες]

Αν έχετε χρησιμοποιήσει κάποιο πρόγραμμα ζωγραφικής στον υπολογιστή (π.χ., Paint), ίσως έχετε κάνει χρήση του "κουβά", του εργαλείου δηλαδή που μας βοηθάει να γεμίζουμε γρήγορα κλειστές επιφάνειες χωρίς εμπόδια. Το Σχήμα 1 δείχνει ένα παράδειγμα:



Σχήμα 1: Οπτικοποίηση της διαδικασίας γεμίσματος με χρώμα. Στο σχήμα στα αριστερά επιλέξαμε την θέση (3, 2) - 3η γραμμή, 2η στήλη - για να την γεμίσουμε με χρώμα. Στο σχήμα στα δεξιά παρατηρούμε το αποτέλεσμα του γεμίσματός μας.

Για τις ανάγκες της εφαρμογής μας κάνουμε τις εξής παραδοχές: (1) έχουμε μόνο δύο χρώματα που αναπαρίστανται με 0 και 1, (2) το πρόγραμμά μας γεμίζει την εικόνα μόνο με το χρώμα 1, (3) οι εικόνες που χειριζόμαστε είναι τετραγωνικές και (4) το χρώμα μπορεί μόνο να γεμίζει επιφάνειες που επικοινωνούν ελεύθερα με το αρχικό σημείο κάνοντας μόνο κινήσεις πάνω, κάτω, αριστερά, δεξιά (όχι διαγώνια). Εάν ένα κελί περιέχει ήδη το χρώμα 1, το κελί δεν θεωρείται ελεύθερο και δρα ως "τοίχος" για το γέμισμά μας.

Γράψτε ένα πρόγραμμα το οποίο παίρνει ως όρισμα το όνομα του αρχείου που περιέχει την εικόνα και το αρχικό σημείο για γέμισμα και τυπώνει στην έξοδο την τελική εικόνα. Η πρώτη σειρά του αρχείου περιέχει την διάσταση της εικόνας, η δεύτερη σειρά τις συντεταγμένες του αρχικού σημείου και τέλος ακολουθεί η δισδιάστατη εικόνα. Παράδειγμα εκτέλεσης ακολουθεί:

```
$ cat picture.txt
6
3 2
000000
001110
010010
010010
011100
000000
$ ./fill picture.txt
000000
001110
011110
011110
011100
000000
```

Απάντηση:

Πρόχειρο