

Προγραμματισμός Ι

Εργαστήριο 8

Διδάσκων: Χρήστος Δίου

Βασισμένο στο υλικό του κ. Δημήτρη Μιχαήλ

1 Μέσος Όρος Πίνακα

Γράψτε μια συνάρτηση που να υπολογίζει τον μέσο όρο των στοιχείων ενός πίνακα. Η συνάρτηση πρέπει να δέχεται δύο παραμέτρους:

1. έναν πίνακα με όνομα `a[]` τύπου `int`,
2. έναν ακέραιο αριθμό `n` που υποδηλώνει το μέγεθος του πίνακα `a[]`.

Ο τύπος επιστροφής της συνάρτησης πρέπει να είναι αριθμός κινητής υποδιαστολής.

Στην συνέχεια γράψτε ένα πρόγραμμα που να γεμίζει έναν πίνακα με 1000 τυχαίους ακέραιους με τιμές μεταξύ του 0 και του 10000 ($0 \leq a[i] \leq 10000$) και να υπολογίζει τον μέσο όρο τους. Μην ξεχάσετε να αρχικοποιήσετε την γεννήτρια ψευδοτυχαίων αριθμών με σπόρο τον χρόνο του συστήματος.

2 Ψευδοτυχαίοι Αριθμοί και Ιστόγραμμα

Θέλουμε να γράψουμε ένα πρόγραμμα που να παράγει 1000000 τυχαίους αριθμούς από το 0 έως και το 99 και να μετράει πόσες φορές εμφανίστηκε ο κάθε αριθμός. Στην συνέχεια πρέπει να εκτυπώνει έναν πίνακα της μορφής:

element	value
0	100
1	203
2	13
.	.

για όλους τους 100 αριθμούς. Χρησιμοποιήστε έναν πίνακα ακεραίων για μετρητές. Μην ξεχάσετε να αρχικοποιήσετε τα στοιχεία του πίνακα σας σε μηδέν και την γεννήτρια ψευδοτυχαίων αριθμών με σπόρο τον χρόνο του συστήματος.

3 Μήκος Συμβολοσειράς

Θυμηθείτε πως οι συμβολοσειρές στην C δεν είναι τίποτα άλλο παρά πίνακες χαρακτήρων. Οι πίνακες όμως αυτοί έχουν το ιδιαίτερο χαρακτηριστικό πως περιέχουν τον χαρακτήρα `'\0'` ο οποίος συμβολίζει το τέλος της συμβολοσειράς.

Γράψτε μια συνάρτηση

```
1 unsigned int mystrlen(const char s[]);
```

η οποία πέρνει ως παράμετρο μια συμβολοσειρά και επιστρέφει το μήκος της. Το μήκος της συμβολοσειράς δεν πρέπει να περιλαμβάνει τον κενό χαρακτήρα `'\0'`.

Για να ελέγξετε τον κώδικα σας χρησιμοποιήστε το παρακάτω πρόγραμμα που διαβάζει μία λέξη από την τυπική είσοδο. Προσοχή πως όταν κάνουμε κλήση της `scanf` και δίνουμε ως είσοδο πίνακα, δεν πρέπει να χρησιμοποιήσουμε τον τελεστή `addressof` `&`.

```
1 int main() {
2     char str[1000];
3
4     printf("Give a word:");
5     scanf("%s", str);
6
7     printf("Length of given word is %d\n", mystrlen(str));
8
9     return 0;
10 }
```

4 Σύγκριση Συμβολοσειρών

Θέλουμε να γράψουμε μια συνάρτηση που να συγκρίνει δύο συμβολοσειρές λεξικογραφικά. Η συνάρτηση πρέπει να πέρνει δύο συμβολοσειρές ως παραμέτρους και να επιστρέφει έναν ακέραιο.

```
1 int mystrcmp(const char s1[], const char s2[]);
```

Στην περίπτωση που η συμβολοσειρά `s1` είναι μικρότερη της `s2` πρέπει να επιστρέφει έναν αρνητικό ακέραιο, εαν είναι ίδιες 0 και έναν θετικό ακέραιο σε περίπτωση που η `s1` είναι μεγαλύτερη λεξικογραφικά της `s2`.

Για την λεξικογραφική σύγκριση ισχύουν τα εξής:

- ο κενός χαρακτήρας είναι πριν από οποιοδήποτε χαρακτήρα, και άρα
`"hello" < "hello world"`
- η σύγκριση των γραμμάτων μπορεί να γίνει απευθείας μέσω του πίνακα ASCII, δηλαδή
`"hEllo" < "hello"`.

Στην συνέχεια γράψτε ένα πρόγραμμα που να διαβάζει δυο συμβολοσειρές από τον χρήστη και να τυπώνει στην οθόνη ποια είναι μεγαλύτερη.

5 Πολυώνυμα

Θέλουμε να γράψουμε ένα μικρό πρόγραμμα που να διαβάζει ένα πολυώνυμο από τον χρήστη, π.χ $p(x) = 3x^4 + 2x^2 + 1$ και μια τιμή x_0 για το x και να υπολογίζει την τιμή του πολυωνύμου $p(x_0)$.

Είσοδος Η είσοδος από τον χρήστη θα είναι της μορφής

$$d \quad a_0 \quad a_1 \quad a_2 \quad \dots \quad a_d$$

όπου d είναι ο βαθμός του πολυωνύμου. Υποθέστε πως ο βαθμός θα είναι πάντα $d \leq 10$ (για να μπορούμε όμως να αλλάξουμε το μέγιστο βαθμό πολυωνύμου, χρησιμοποιήστε μια σταθερά του προ-επεξεργαστή για να συμβολίσετε τον μέγιστο βαθμό). Η είσοδος αυτή αντιστοιχεί στον πολυώνυμο

$$p(x) = a_d \cdot x^d + \dots + a_2 \cdot x^2 + a_1 \cdot x + a_0.$$

Το πολυώνυμο αυτό μπορεί να αποθηκευτεί σε ένα πίνακα κατάλληλου μεγέθους, όπου στην θέση i του πίνακα έχουμε αποθηκεύσει τον συντελεστή a_i .

Ο χρήστης εκτός από το πολυώνυμο θα δίνει αμέσως μετά μια τιμή ακόμη που θα είναι η τιμή x_0 για την οποία θέλουμε να υπολογίσουμε την τιμή του πολυωνύμου $p(x_0)$. Η έξοδος του προγράμματος σας πρέπει να είναι η τιμή $p(x_0)$.

Παράδειγμα εισόδου

3 1 5 0 3
2

που αντιστοιχεί στο πολυώνυμο $p(x) = 1 + 5 \cdot x + 3 \cdot x^3$ βαθμού 3. Η τιμή του πολυωνύμου για $x_0 = 2$ είναι $p(2) = 35$ και άρα το πρόγραμμα σας πρέπει να τυπώσει 35.

Αλγόριθμος Υπολογισμού Ένα πολυώνυμο μπορεί να υπολογιστεί αποδοτικά για μια τιμή του x χρησιμοποιώντας το σχήμα **Horner**. Το σχήμα Horner μας λέει πως μπορούμε να υπολογίσουμε αποδοτικά ένα πολυώνυμο εάν γράψουμε το πολυώνυμο ως

$$p(x) = a_d \cdot x^d + \dots + a_2 \cdot x^2 + a_1 \cdot x + a_0 = ((a_d \cdot x + a_{d-1}) \cdot x + \dots) \cdot x + a_0.$$

Ο αλγόριθμος σας πρέπει να κάνει λοιπόν τα εξής:

1. να αρχικοποιήσει μια μεταβλητή `t` με την τιμή a_d
2. για κάθε a_i από $i = d - 1$ έως και $i = 0$ να πολλαπλασιάζει την μεταβλητή `t` με x_0 και μετά να προσθέτει a_i .

Στο τέλος του αλγορίθμου η μεταβλητή `t` θα περιέχει την τιμή του πολυωνύμου στο σημείο x_0 , δηλαδή την τιμή $p(x_0)$.