

Προγραμματισμός Ι

Εργαστήριο 7

Διδάσκων: Χρήστος Δίου

Βασισμένο στο υλικό του κ. Δημήτρη Μιχαήλ

1 Ψευδοτυχαίοι Αριθμοί

Η γλώσσα C μας παρέχει στην βιβλιοθήκη της μια συνάρτηση για την παραγωγή αριθμών που μοιάζουν τυχαίοι. Η συνάρτηση ονομάζεται `rand()` και κάθε φορά που καλείτε παράγει έναν ακέραιο ανάμεσα στο 0 και το `RAND_MAX`. Η συνάρτηση αυτή και η σταθερά βρίσκονται στο αρχείο `<stdlib.h>`.

1.1 Γεννήτρια

Γράψτε ένα πρόγραμμα που να τυπώνει 20 τυχαίους αριθμούς μεταξύ 0 και 999.

1.2 Σπόρος

Εκτελέστε το προηγούμενο πρόγραμμα πολλές φορές. Τι παρατηρείτε; Είναι οι αριθμοί διαφορετικοί ή ίδιοι;

Οι αριθμοί που παράγονται από την `rand()` δεν είναι πραγματικά τυχαίοι, είναι μια ακολουθία αριθμών που μοιάζουν τυχαίοι αλλά επαναλαμβάνονται. Ο πρώτος αριθμός της ακολουθίας μπορεί να επηρεαστεί από την συνάρτηση `srand()` η οποία παρέχει ένα "σπόρο" (seed) στην `rand()`. Η αρχική τιμή του σπόρου είναι πάντα 1. Η συνάρτηση `rand()` με ίδιο σπόρο, παράγει τους ίδιους αριθμούς.

Στην αρχή του προγράμματος της προηγούμενης ενότητας, ζητήστε από τον χρήστη έναν αριθμό μεταξύ 0 και 100. Καλέστε στην συνέχεια την `srand()` με αυτόν τον αριθμό ως παράμετρο. Εκτελέστε το πρόγραμμα μερικές φορές ακόμη. Τι παρατηρείτε;

1.3 Χρόνος για Σπόρος

Μια συχνή τακτική που χρησιμοποιούν οι προγραμματιστές είναι να καλούν την `srand()` με σπόρο έναν αριθμό που να εξαρτάται από τον χρόνο στο σύστημα. Με αυτό τον τρόπο μπορούν να υποθέσουν πως θα είναι μικρή η πιθανότητα να χρησιμοποιείται συνέχεια ο ίδιος σπόρος.

Για να μην παράγονται πάντα οι ίδιοι τυχαίοι αριθμοί κάντε

```
1 #include <time.h>
```

και καλέστε την `srand()` με σπόρο:

```
1 srand(time(NULL));
```

Η συνάρτηση `time()` γυρνάει τον αριθμό των δευτερολέπτων που έχουν περάσει από την χρονική στιγμή: "00:00:00 UTC, January 1, 1970" και είναι μια συνήθης επιλογή για σπόρο σε γεννήτρια ψευδοτυχαίων αριθμών.

2 Κέρμα

Γράψτε ένα πρόγραμμα που να ζητάει από τον χρήστη έναν αριθμό $0 < n < 1000$, να ρίχνει ένα κέρμα n φορές και να μετράει τον αριθμό των φορών που ήρθε κορώνα ή γράμματα. Το πρόγραμμα σας πρέπει να τυπώνει π.χ:

```
Give number of times:
100
Throwing coin 100 times:
heads: 53
tails: 47
```

Φροντίστε να αρχικοποιήσετε την γεννήτρια τυχαίων αριθμών με τον χρόνο του συστήματος. Μοντελοποιήστε την ρίψη του κέρματος με μια συνάρτηση που θα γυρνάει `true` ή `false` ανάλογα με το αποτέλεσμα την ρίψης. Ονομάστε αυτή την συνάρτηση `coin_toss()`.

3 Πίνακες και Παραγοντικό

Υλοποιήστε το παρακάτω πρόγραμμα ώστε να γεμίζει τον πίνακα `factorial` με τα παραγοντικά από το 0! έως και το 14!. Το $i!$ πρέπει να αποθηκευτεί στην θέση i του πίνακα. Στην συνέχεια τυπώστε τις τιμές ξεκινώντας από το 14! έως και το 0! ανά γραμμή.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main ()
5  {
6      long factorial[15];
7
8      // here compute 0! up to 14!
9
10     // here print array factorial
11 }
```

4 Μέγιστο Στοιχείο Πίνακα

Γράψτε μια συνάρτηση που να επιστρέφει τον μέγιστο ενός πίνακα ακεραίων. Η συνάρτηση πρέπει να δέχεται δύο παραμέτρους:

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

Στην συνέχεια γράψτε ένα πρόγραμμα που να γεμίζει έναν πίνακα με 1000 τυχαίους αριθμούς και να εκτυπώνει τον μέγιστο αυτών. Μη ξεχάσετε να αρχικοποιήσετε την γεννήτρια ψευδοτυχαίων αριθμών με σπόρο τον χρόνο του συστήματος.