

Računske vježbe 6

Programiranje I

1. Napisati program koji simulira bacanje kocke i koji provjerava da li je kocka fer. Kocka je fer ukoliko su svi ishodi bacanja kocke (1, 2 ..., 6) približno vjerovatni.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int count[6] = {0}; // inicijalizujemo sve članove niza na 0
7     int i, dice, repeat = 10000;
8
9     srand(time(NULL));
10
11     for (i = 0; i < repeat; i++)
12     {
13         dice = throw_dice();
14         count[dice - 1]++; //throw_dice() daje 1-6, a indeksi su 0-5
15     }
16
17     printf("Vjerovatnoce su: ");
18     for (i = 0; i < 6; i++)
19     {
20         printf("%lf ", count[i] / (double)repeat);
21     }
22     printf("\nOcekivana vjerovatnoca je: %lf", 1.0 / 6);
23 }
24
25 int throw_dice()
26 {
27     return 1 + rand() % 6; // sa 1 opseg 0-5 pomjeramo kako bismo dobili opseg 1-6
28 }
```

Mada bi nam Code::Blocks IDE (integrisano razvojno okruženje (engl. *integrated development environment*)) sve do sada dodavao zaglavlje `stdlib.h` kada napravimo novi projekat mi smo ga brisali jer nam nije bilo potrebno. Ovo je zaglavlje za **Standard Library** biblioteku koja nam pruža razne korisne funkcije i sa nekim od njih ćemo se upoznati u toku kursa. Funkcija `rand()` koju nam pruža standardna biblioteka predstavlja pseudo-slučajni generator brojeva. Sekvenca slučajnih brojeva generisana ovim generatorom nije stvarno slučajna jer je u potpunosti određena u zavisnosti od sjemena (engl. **seed**) koje predstavlja inicijalnu vrijednost. Dakle, ako znamo sjeme nečijeg generatora možemo ponoviti istu sekvencu. Da bi se što više približili nasumičnim brojevima obično kao sjeme koristimo tekuće vrijeme u sekundama. Funkcija `time()` računa razliku od Unix epohe (referentna tačka u računarskom vremenu: 00:00:00 UTC on 1 January 1970) do datog trenutka u sekundama. Kao argument uzima pokazivač u koji smješta rezultat mada ga i vraća. Mi joj proslijedujemo `NULL` pokazivač jer nam rezultat dalje neće biti potreban. Funkcija `srand()` „sije” generator nasumičnih brojeva. Opseg funkcije `rand()` možemo kontrolisati pomoću operatora `%`:

```
broj = rand() % duzina;
```

jer znamo da ostatak pri dijeljenju može biti u intervalu od 0 do `duzina-1`.

2. Napisati program koji miješa niz X. U tu svrhu, generisati nasumičnu permutaciju indeksa niza X (npr. od redosljeda indeksa 0, 1, ..., N-1, pravi redosljed 5, 9, 0, ..., 4), pa presložiti elemente u skladu sa novim redosljedom indeksa.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void random_indices(int *, int);
5 void shuffle(int *, int *, int);
6
7 int main()
8 {
9     int indices[10] = {0};
10    int x[10] = {11, 22, 23, 38, 49, 55, 101, 103, 177, 202};
11    // Pokrenite program bez ove naredbe nekoliko puta. Sta uocavate?
12    srand(time(NULL));
13
14    puts("Niz je: ");
15    for (int i = 0; i < 10; i++)
16        printf("%d ", x[i]);
17
18    random_indices(indices, 10);
19    puts("\nIndeksi su: ");
20    for (int i = 0; i < 10; i++)
21        printf("%d ", indices[i]);
22
23    shuffle(x, indices, 10);
24    puts("\nPreslozeni niz je: ");
25    for (int i = 0; i < 10; i++)
26        printf("%d ", x[i]);
27 }
28
29 void random_indices(int *x, int length)
30 {
31     int i, t, index;
32
33     for(i = 0; i < length; i++)
34         x[i] = i; // redom upisemo sve moguće vrijednosti indeksa
35
36     // vrsimo zamjenu elemenata na tekućoj poziciji i nasumičnoj poziciji index.
37     for(i = 0; i < length; i++)
38     {
39         index = rand() % length; // generisemo nasumican indeks
40         t = x[i];
41         x[i] = x[index];
42         x[index] = t;
43     }
44 }
45
46 void shuffle(int *arr, int *indices, int length)
47 {
48     int i;
49     int temp[length]; // pomocni niz
50     for (i = 0; i < length; i++)
51         temp[i] = arr[indices[i]]; // vrsimo upis po novom redosljedu
52     for (i = 0; i < length; i++)
53         arr[i] = temp[i]; // slozimo nazad u odredisni niz
54 }
```