

Pseudo-slučajni brojevi i dinamičko zauzimanje memorije

Laboratorijska vježba 7

Uvod

U nastavku su navedeni zadaci koje je potrebno riješiti. Pri rješavanju paziti na odabir prikladnih ili zadane tipove podataka. U potpunosti rukovati memorijom na hrpi (zauzimanje memorije, provjera uspješnosti zauzimanja i oslobađanje memorije). Inicijalizirati na uobičajeni način generator pseudo-slučajnih brojeva.

Zadaci

1. Pronaći i ispraviti greške u sljedećem tekstu programa.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void)
5  {
6      int n;
7      double p[], sum = 0;
8
9      do { scanf("%d", n); } while (n < 5);
10
11     p = (double)malloc(n * sizeof(double*));
12     if (p != NULL)
13         return 1;
14
15     for (i = 0; i < n; i++) {
16         *p[i] = (double)rand() / RAND_MAX - 1;
17         sum += p[i];
18     }
19
20     for (i = 0; i < n; i++)
21         if ((p+i) > sum / n)
22             printf("%f\n", *(p+i))
23
24     free(p);
25     return 0;
26 }
```

2. Napisati funkciju koja u predanom joj stringu postavlja sve znakove koji nisu slova u pseudo-slučajno generirano veliko ili malo slovo. U svrhu testiranja u funkciji `main()` dinamički zauzeti memoriju za 50 podataka tipa `char` (u potpunosti rukovati zauzetom memorijom). Omogućiti korisniku unos stringa u dobiveno

polje (osigurati da se ne premaši njegova veličina) te ga iskoristi u pozivu napisane funkcije. Naknadno ispisati string na ekran.

3. Napisati funkciju koja predano joj polje realnih brojeva popunjava pseudo-slučajnim brojevima iz $[-2n, 2n] \setminus [-n, 0]$ (vrijednosti iz $[-n, 0]$ nisu dozvoljene), gdje je n veličina polja. U svrhu testiranja u funkciji `main()` dinamički zauzeti memoriju za 800 podataka tipa `float` (u potpunosti rukovati zauzetom memorijom). Iskoristiti dobiveno polje pri pozivu napisane funkcije i naknadno ispisati elemente polja na ekran.
4. Omogućiti korisniku unos cijelog broja $5 < n \leq 20$. Potom, n puta generirati pseudo-slučajan broj t iz $[10, 1000] \subset \mathbb{Z}$ te dinamički zauzeti memoriju za t podataka tipa `double`. Svaki puta dobiveno polje popuniti pseudo-slučajnim brojevima iz $[-10, 0] \subset \mathbb{R}$, pronaći najmanji i najveći elemenat te ispisati apsolutnu vrijednost njihove razlike na ekran. U potpunosti rukovati zauzetom memorijom.
5. Generirati pseudo-slučajni broj iz $[1000, 10000]$ te dinamički zauzeti memoriju za toliko podataka tipa `double` (u potpunosti rukovati zauzetom memorijom). Popuniti dobiveno polje pseudo-slučajnim brojevima iz $[-5, 20] \subset \mathbb{R}$. Izračunati aritmetičku sredinu i pronaći element polja koji joj je najbliži.
6. Napisati funkciju koja generira i vraća cijeli broj iz $[\gamma, \delta]$ koji je neparan. U svrhu testiranja u funkciji `main()` pozvati napisanu funkciju 128 puta s -2368 i 2998 kao argumentima te svaki put na ekran ispisati povratnu vrijednost.
7. Napisati funkciju koja u predanom joj stringu mijenja sve znakove u pseudo-slučajno generirane znamenke. U svrhu testiranja u funkciji `main()` generirati pseudo-slučajni broj $10 < n < 111$ te dinamički zauzeti memoriju za toliko podataka tipa `char` (u potpunosti rukovati zauzetom memorijom). Omogućiti korisniku unos stringa u dobiveno polje (paziti da se ne premaši veličina polja) te ga iskoristiti u pozivu napisane funkcije. Na kraju, ispisati izmijenjeni string na ekran.
8. Napisati funkciju koja u predano joj polje cijelih brojeva popunjava pseudo-slučajnim parnim brojevima iz $[100, 10000]$. U svrhu testiranja u funkciji `main()` dinamički zauzeti memoriju za 200 podataka tipa `long int` (u potpunosti rukovati zauzetom memorijom). Iskoristiti dobiveno polje u pozivu napisane funkcije i na ekran ispisati popunjeno polje.
9. Napisati funkciju koja predano joj polje cijelih brojeva popunjava pseudo-slučajnim brojevima djeljivim sa sedam iz $[2000, 30000] \subset \mathbb{Z}$. Izdvojiti korak provjere generiranja pseudo-slučajnih brojeva djeljivih sa sedam u posebnu funkciju. U svrhu testiranja u funkciji `main()` dinamički zauzeti memoriju za 150 podataka tipa `int` (u potpunosti rukovati zauzetom memorijom). Iskoristiti dobiveno polje u pozivu napisane funkcije i naknadno ispisati elemente polja na ekran.

10. Napisati funkciju koja predano joj polje znakova popunjava pseudo-slučajno generiranim slovima te na kraj (zadnji element polja) postavlja *null*-znak kako bi se dobio string. U svrhu testiranja u funkciji `main()` dinamički zauzeti memoriju za 36 podataka tipa `char` (u potpunosti rukovati zauzetom memorijom). Iskoristiti dobiveno polje u pozivu napisne funkcije te ispisati dobiveni string na ekran.
11. Omogućiti korisniku unos cijelog broja $9 < n \leq 200$. Potom, dinamički zauzeti memoriju n podataka tipa `double` te popuniti tako dobiveno polje pseudo-slučajnim brojevima iz $[-300, 300]$. Nasumično odabrati $\lfloor n/2 \rfloor$ elementa (nasumično generirati indekse elemenata) te izračunati i na ekran ispisati njihovu aritmetičku sredinu.