

Pokazivači i dinamičko zauzimanje memorije

Laboratorijska vježba 7

Uvod

U nastavku su navedeni zadaci koje je potrebno riješiti. Pri rješavanju paziti na prikladnost odabranih tipova podataka, kao i na nužne veličine polja. U potpunosti rukovati dinamički zauzetom memorijom na hrpi (zauzimanje memorije, provjera uspješnosti zauzimanja i oslobađanje memorije).

Zadaci

1. Pronaći i ispraviti greške u sljedećem tekstu programa.

```
1  #include <stdio.h>
2
3  double average(int nums*, int n);
4
5  int main(void)
6  {
7      double seq[] = {1.2, 2.3, 3.4, 4.5, 5.6, 6.7, 7.8, 8.9, 9.1};
8      double avg;
9
10     avg = average(*seq, 9);
11     printf("%f\n", avg);
12
13     return 0;
14 }
15
16 double average(int []nums, int n) {
17     int sum = 0, i;
18     for (i = 1; i <= n; i++) {
19         sum += (nums + i);
20     }
21     return (double)sum / n;
22 }
```

2. Deklarirati i inicijalizirati cjelobrojnu varijablu te pokazivač na nju. Ispisati joj vrijednost i adresu. Ukoliko je pozitivna, udvostručiti joj vrijednost, a u suprotnom je prepoloviti te joj ponovo ispisati vrijednost i adresu. Rukovati s vrijednosti varijable isključivo posredstvom pokazivača.
3. Omogućiti korisniku unos $4 \leq m < 19$ vrijednosti u polje cijelih brojeva. Odrediti i na ekran ispisati razliku elementa s najvećom i elementa s najmanjom vrijednosti. Koristiti pokazivačku notaciju za pristup elementima polja.

4. Napisati funkciju koja računa i vraća

$$f(\mathbf{x}) = x_1 \cdot \sum_{i=1}^{n-1} |i \cdot x_{i+1}^2 - 1|, \quad \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n.$$

U svrhu testiranja u funkciji `main()` deklarirati polje od 10 elemenata i naknadno omogućiti korisniku njegovo popunjavanje. Iskoristiti to polje u pozivu navedene funkcije i ispisati na ekran povratnu vrijednost.

5. Omogućiti korisniku unos $3 < m \leq 18$ elemenata u polje cijelih brojeva. Odrediti i na ekran ispisati broj s najvećim zbrojem znamenki. Koristiti pokazivačku notaciju za pristup elementima polja.
6. Napisati funkciju koja vraća vrijednost -1 ukoliko predani joj string sadrži više malih slova u odnosu na velika, a u obratnom slučaju vraća vrijednost 1 . Ukoliko string ne sadrži niti jedno slovo ili sadrži jednak broj malih i velikih slova funkcija vraća vrijednost 0 . U svrhu testiranja u funkciji `main()` omogućiti korisniku unos stringa od maksimalno 100 znakova u prethodno deklarirano polje odgovarajuće veličine. Iskoristiti navedeni string u pozivu napisane funkcije i na ekran ispisati povratnu vrijednost. Koristiti isključivo pokazivačku notaciju za pristup elementima polja.
7. Napisati funkciju koja predani joj string pretvara u odgovarajući cijeli broj kojeg vraća. Ukoliko string ne predstavlja slijed znakova koji predstavlja cijeli broj, funkcija vraća vrijednost `INT_MAX`. Primjerice, za string `"-1234"`, funkcija bi trebala vratiti cjelobrojni vrijednost -1234 . U svrhu testiranja u funkciji `main()` omogućiti korisniku unos stringa od maksimalno 6 znakova u prethodno deklarirano polje odgovarajuće veličine. Iskoristiti navedeni string u pozivu napisane funkcije i na ekran ispisati povratnu vrijednost.
8. Napisati funkciju koja u predanom joj polju realnih brojeva zamjenjuje mjesta elementima prve polovice s elementima druge polovice i to na način da se zamjeni prvi i zadnji element, zatim drugi i predzadnji i tako dalje. U svrhu testiranja u funkciji `main()` omogućiti korisniku unos $5 < n < 11$ elemenata u polje realnih brojeva. Iskoristiti popunjeno polje u pozivu napisane funkcije te ispisati izmijenjeno polje na ekran.
9. Napisati funkciju koja elemente predanog joj polja cijelih brojeva mijenja na sljedeći način: elemente manje od aritmetičke sredine svih elemenata polja zamjenjuje s tom istom aritmetičkom sredinom. Korak izračuna aritmetičke sredine elemenata polja izdvojiti u zasebnu funkciju. U svrhu testiranja u funkciji `main()` omogućiti korisniku unos $3 < n \leq 16$ elemenata u polje cijelih brojeva. Iskoristiti popunjeno polje u pozivu napisane funkcije te ispisati izmijenjeno polje na ekran. Koristiti isključivo pokazivačku notaciju za pristup elementima polja.

10. Napisati funkciju koja računa i vraća zbroj ASCII vrijednosti znakova prednog joj stringa. U svrhu testiranja u funkciji `main()` omogućiti korisniku unos stringa od maksimalno 60 znakova u prethodno deklarirano polje odgovarajuće veličine. Potom, iskoristiti navedni string u pozivu napisane funkcije te ispisati na ekran odgovarajuću poruku je li dobiveni zbroj djeljiv s duljinom stringa ili nije.
11. 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.
12. 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.
13. 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 element te ispisati apsolutnu vrijednost njihove razlike na ekran. U potpunosti rukovati zauzetom memorijom.
14. 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.
15. 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 bez ponavljanja) te izračunati i na ekran ispisati njihovu aritmetičku sredinu.
16. Napisati funkciju koja računa i vraća

$$f(\mathbf{x}) = \frac{1}{n} \cdot \sum_{i=2}^{n-1} (|x_{i-1} - x_{i+1}^2| + x_i^3), \quad \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n.$$

U svrhu testiranja u funkciji `main()` deklarirati polje od 10 elemenata i naknadno omogućiti korisniku njegovo popunjavanje. Iskoristiti to polje u pozivu navedene funkcije i ispisati na ekran povratnu vrijednost.

17. 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 }
```
