# Programski jezik I

– Primjeri sa 7. termina predavanja –

## Pretraživanje, sortiranje

1. Napisati program koji sadrži funkciju `brute_force_pretraga` koja u nizu cijelih brojeva `x` koji joj se prosljeđuje kao parametar (zajedno sa dužinom) traži element koji je funkciji proslijeđen kao treći parametar. U slučaju da funkcija pronađe traženi elemenat, vraća njegovu poziciji, a u suprotnom vraća -1. Korisnik unosi u glavnom programu niz i cio broj `T`, a zatim se poziva funkcija i štampa rezultat.

   **Primjer:** Ako se unese niz `x = [12 2 8 3 6 5 11 100 13 17]` i `T = 100`, funkcija vraća 7.

   **Rješenje:**

```c
#include <stdio.h>

int brute_force_pretraga(int *, int, int);

int main()
{
        int arr[] = {12, 2, 8, 3, 6, 5, 11, 100, 13, 17};

        int n = 10;
        int T = 100;
        int pozicija = brute_force_pretraga(arr, n, T);
        if (pozicija != -1)
        {
                printf("Element se nalazi na indeksu %d", pozicija);
        }
        else
        {
                printf("Element nije prisutan u nizu.");
        }
}

// Funkcija koja vrši pretragu niza pomoću brute force metode
int brute_force_pretraga(int *arr, int n, int T)
{
        for (int i = 0; i < n; i++)
        {
                if (arr[i] == T)
                {
                        return i; // vraćamo indeks pronađenog elementa
                }
        }
        return -1; // varaćamo -1 ako element nije pronađen
}
```

2. Napisati program koji sadrži funkciju **`binarna_pretraga`** koja sortiranom nizu cijelih brojeva **`x`** koji joj se prosljeđuje kao parametar (zajedno sa dužinom) traži element koji je funkciji proslijeđen kao treći parametar. U slučaju da funkcija pronađe traženi elemenat, vraća njegovu poziciji, a u suprotnom vraća -1. Korisnik unosi u glavnom programu niz i cio broj **`T`**, a zatim se poziva funkcija i štampa rezultat.

   **Primjer:** Ako se unese niz **`x = [12 2 8 3 6 5 11 100 13 17]`** i **`T = 100`**, funkcija vraća 7.

   **Rješenje:**

```c
#include <stdio.h>

int binarna_pretraga(int[], int, int);
int main()
{

        // pozdrav();
        int arr[] = {1, 2, 8, 13, 26, 35, 41, 100, 113, 117};
        int n = 10;

        int T = 100;
        int pozicija = binarna_pretraga(arr, n, T);
        if (pozicija != -1)
        {
                printf("Element se nalazi na indeksu %d", pozicija);
        }
        else
        {
                printf("Element nije prisutan u nizu.");
        }
}

int binarna_pretraga(int niz[], int velicina, int T)
{
        int lijevo = 0;
        int desno = velicina - 1;
        int sredina;
        while (lijevo <= desno)
        {
                sredina = lijevo + (desno - lijevo) / 2;
                if (niz[sredina] == T)
                return sredina;
                if (niz[sredina] < T)
                lijevo = sredina + 1;
                else
                desno = sredina - 1;
        }
        return -1; // Ako se element ne nađe, vraćamo -1
}
```

3. Napisati program koji sadrži funkciju **ponovljeni_minimum** koja vrši sortiranje niza cijelih brojeva **x** koji joj se prosljeđuje kao parametar (zajedno sa dužinom). Korisnik unosi u glavnom programu niz, a zatim poziva realizovanu funkciju i štampa sortirani niz.

**Rješenje:**

```
01 | #include <stdio.h>
02 |
03 | void ponovljeni_minimum(int *, int);
04 | void ponovljeni_minimum_varijanta2(int *, int);
05 | void bubble_sort(int *, int);
06 |
07 | int main()
08 | {
09 |         int arr[] = {12, 2, 8, 3, 6, 5, 11, 100, 13, 17};
10 |
11 |         int i, n = 10;
12 |
13 |         bubble_sort(arr, n);
14 |
15 |         for (i = 0; i < n; i++)
16 |         {
17 |                 printf("%d ", arr[i]);
18 |         }
19 | }
20 |
21 | // Funkcija koja vrši sortiranje niza metodom ponovljenog  minimuma
22 | void ponovljeni_minimum(int *arr, int n)
23 | {
24 |         int i, j, min, min_index, temp;
25 |         for (i = 0; i < n - 1; i++)
26 |         {
27 |                 for (j = i + 1; j < n; j++)
28 |                 if (arr[i] > arr[j])
29 |                 {
30 |                         temp = arr[i];
31 |                         arr[i] = arr[j];
32 |                         arr[j] = temp;
33 |                 }
34 |         }
35 | }
36 |
37 | // Funkcija koja vrši sortiranje niza metodom ponovljenog  minimuma
      unaprijeđena varijanta
38 | void ponovljeni_minimum_varijanta2(int *arr, int n)
39 | {
40 |         int i, j, min, min_index, temp;
41 |         for (i = 0; i < n - 1; i++)
42 |         {
43 |                 min_index = i;
44 |                 for (j = i + 1; j < n; j++)
45 |                 {
46 |                         if (arr[min_index] > arr[j])
47 |                         {
48 |                                 min_index = j;
49 |                         }
50 |                 }
51 |                 temp = arr[i];
52 |                 arr[i] = arr[min_index];
53 |                 arr[min_index] = temp;
54 |         }
55 | }
56 |
```

```
57 | // Funkcija koja vrši sortiranje niza metodom bubble sort
58 | void bubble_sort(int *arr, int n)
59 | {
60 |         int i, j, min, min_index, temp, ind;
61 |         for (i = n - 1; i > 0; i--)
62 |         {
63 |                 ind = 1; // ind=1 nije došlo do zamjene elemenata; ind=0
       jeste
64 |                 for (j = 0; j < i; j++)
65 |                 {
66 |                         if (arr[j] > arr[j + 1])
67 |                         {
68 |                                 temp = arr[j];
69 |                                 arr[j] = arr[j + 1];
70 |                                 arr[j + 1] = temp;
71 |                                 ind = 0; // Došlo je do zamjene elemenata
72 |                         }
73 |                 }
74 |                 if (ind == 1)
75 |                 {
76 |                         break;
77 |                 }
78 |         }
79 | }
```

4. Napisati program koji poziva funkciju definisanu u drugom fajlu. Demonstrirati `#define` direktivu.

   **Rješenje:**

   Fajl `funkcije.h`

```
01 | #include <stdio.h>
02 | #define a 5
03 | void dobrodosli()
04 | {
05 |         printf("Dobrodosli!\n");
06 | }
07 |
08 | void stampaj()
09 | {
10 |         printf("Demonstracija direktive #define. štampamo njenu vrijednost
       %d!\n", a);
11 | }
```

   Glavni program

```
01 |         #include <stdio.h>
02 |         #include "funkcije.h"
03 |         int main()
04 |         {
05 |                 dobrodosli();
06 |                 stampaj();
07 |         }
08 |
```