

PROGRAMIRANJE 1

PROGRAMSKI JEZIK

C

Staša Vujičić
Čas 10

FUNKCIJE. GLOBALNE PROMENLJIVE.

- ◉ Brojač karaktera

Program čita tekst sa ulaza i broji velika i mala slova, cifre, beline i redove. Program demonstrira korišćenje globalnih podataka.

```
#include <stdio.h>
```

```
/* Deklaracija funkcije prebroj */
```

```
void prebroj();
```

```
/* Globalni brojaci. Podrazumevano su  
inicijalizovani nulom. */
```

```
int br_malih,
```

```
br_velikih,
```

```
br_cifara,
```

```
br_belina,
```

```
br_redova;
```

```
/* main funkcija */
int main()
{

/* Pozivamo funkciju za prebrojavanje */
prebroj();

/* Prikazujemo vrednosti globalnih brojaca,
   koje je funkcija prebroj() izmenila */
printf("Broj malih slova: %d\n", br_malih);
printf("Broj velikih slova: %d\n", br_velikih);
printf("Broj cifara: %d\n", br_cifara);
printf("Broj belina: %d\n", br_belina);
printf("Broj redova: %d\n", br_redova);
}
```

```
/* Definicija funkcije prebroj */  
void prebroj()  
{  
    int c;  
  
    while((c = getchar()) != EOF)  
    {  
        if(c >= 'a' && c <= 'z')    br_malih++;  
        else if(c >= 'A' && c <= 'Z')    br_velikih++;  
        else if(c >= '0' && c <= '9')    br_cifara++;  
        else if(c == '\n' || c == '\t' || c == ' ' )  
        {  
            br_belina++;  
            if(c == '\n')  
                br_redova++;  
        }  
    }  
}
```

POKAZIVAČI

- PRIMER 1 -

- ◉ Zamena promenljivih

Program unosi vrednosti dve promenljive i zatim u posebnoj funkciji zamenjuje njihove vrednosti.

Program demonstrira neispravnu i ispravnu verziju ove funkcije (primenom pokazivača).

```
#include <stdio.h>
```

```
/* Ova funkcija zamenjuje vrednosti  
   lokalnih podataka a i b, sto nije  
   ono sto zelimo da uradimo */
```

```
void swap1(int a, int b)
```

```
{
```

```
    int t;
```

```
    t = a;
```

```
    a = b;
```

```
    b = t;
```

```
}
```

/* Ova funkcija zamenjuje vrednosti
promenljivih na koje pokazuju
lokalne pokazivacke promenljive
a i b. Ovo je upravo ono sto zelimo */

```
void swap2(int *a, int *b)
```

```
{
```

```
    int t;
```

```
    t = *a;
```

```
    *a = *b;
```

```
    *b = t;
```

```
}
```



```
/* main funkcija */
int main()
{
    int x, y;

    printf("Uneti X i Y: ");
    scanf("%d%d", &x, &y);

    printf("X: %d\nY: %d\n", x, y);
    /* X: 2
       Y: 4 */
    swap1(x,y); /* NE RADI */
    printf("Nakon poziva swap1(x,y):\n");
    printf("X: %d\nY: %d\n", x, y);
```

```
/* Nakon poziva swap1(x,y):
```

```
  X: 2
```

```
  Y: 4 */
```

```
swap2(&x,&y); /* RADI */
```

```
  printf("Nakon poziva swap2(x,y):\n");
```

```
  printf("X: %d\nY: %d\n", x, y);
```

```
}
```

```
/* Nakon poziva swap2(x,y):
```

```
  X: 4
```

```
  Y: 2 */
```

POKAZIVAČI

- PRIMER 2 -

- Brojanje karaktera

Program učitava tekst sa ulaza i broji mala i velika slova, cifre, beline i redove. Program demonstrira prenos parametara po adresi, primenom pokazivača.

```
#include <stdio.h>
```

```
/* Deklaracija funkcije prebroj */  
void prebroj(int * br_malih, int * br_velikih,  
            int * br_cifara, int * br_belina,  
            int * br_redova);
```

```
/* main funkcija */  
int main()  
{  
int br_malih = 0,  
  br_velikih = 0,  
  br_cifara = 0,  
  br_belina = 0,  
  br_redova = 0;
```

```
/* Pozivamo funkciju za prebrojavanje */  
prebroj(&br_malih, &br_velikih,  
        &br_cifara, &br_belina,  
        &br_redova);
```

```
/* Prikazujemo vrednosti globalnih brojaca,  
   koje je funkcija prebroj() izmenila */  
printf("Broj malih slova: %d\n", br_malih);  
printf("Broj velikih slova: %d\n", br_velikih);  
printf("Broj cifara: %d\n", br_cifara);  
printf("Broj belina: %d\n", br_belina);  
printf("Broj redova: %d\n", br_redova);  
}
```

```
/* Definicija funkcije prebroj */
void prebroj(int * br_malih, int * br_velikih, int * br_cifara,
             int * br_belina, int * br_redova)
{
    int c;

    while((c = getchar()) != EOF)
    {
        if(c >= 'a' && c <= 'z')    (*br_malih)++;
        else if(c >= 'A' && c <= 'Z')    (*br_velikih)++;
        else if(c >= '0' && c <= '9')    (*br_cifara)++;
        else if(c == '\n' || c == '\t' || c == ' ')
        {
            (*br_belina)++;
            if(c == '\n')
                (*br_redova)++;
        }
    }
}
```

NIZOVI — OSNOVNI POJMOVI

- Niz predstavlja kolekciju elemenata istog tipa. Primer deklaracije niza je:
- **int niz[5];**
/* niz od 5 elemenata tipa int*/

NIZOVI — OSNOVNI POJMOVI

- Pristupanje elementima niza se ostvaruje na sledeći način:
- $niz[0] = 4;$
- $niz[1] = 2 * niz[0]; /*niz[1] = 8*/$
- $niz[2] = niz[0] * niz[1]; /*niz[2] = 32*/$
- $niz[3] = 5;$
- $niz[4] = 7;$
- Napomena: Prvi element u nizu se označava indeksom 0 a ne 1!

NIZOVI — OSNOVNI POJMOVI

- ◉ Indeks niza može da bude proizvoljan izraz celobrojne vrednosti. Na primer može se napisati
 $niz[i*2]=5$.

UNOS VREDNOSTI ELEMENATA NIZA

- Primer unosa vrednosti elemenata niza sa standardnog ulaza:

```
for(i=0; i<5; i++)  
    scanf("%d ", &a[i]);
```

ŠTAMPANJE ELEMENATA NIZA

- ◉ Štampanje elemenata niza na standardni izlaz.

```
for(i=0; i<5; i++)  
    printf("%d ", a[i]);
```

NIZOVI

- PRIMER 1 -

- Program ilustruje korišćenje statičkih nizova. Ispisuje 10 unetih brojeva unazad.

```
#include <stdio.h>
main()
{
int a[10];
int i;
```

```
for (i = 0; i<10; i++)  
{  
    printf("a[%d]=",i);  
    scanf("%d",&a[i]);  
}
```

```
printf("Unazad : \n");
```

```
for (i = 9; i>=0; i--)  
    printf("a[%d]=%d\n",i,a[i]);  
}
```

NIZOVI

- PRIMER 2 -

- Napisati program koji vrši brojanje pojavljivanja svake od cifara na standardnom ulazu.

Koristiti niz brojača.

```
#include <stdio.h>
#include <ctype.h>
main()
{
    /* Niz brojaca za svaku od cifara */
    int br_cifara[10];
    int i, c;
```

```
/* Resetovanje brojaca */
for (i = 0; i < 10; i++)
    br_cifara[i] = 0;

/* Citamo sa ulaza i povecavamo odgovarajuce
   brojace */
while ((c = getchar()) != EOF)
    if (isdigit(c)) /* moze i if(c>='0'&& c<='9') */
        br_cifara[c-'0']++;
/* Ispis rezultata */
for (i = 0; i < 10; i++)
    printf("Cifra %d se pojavila %d put%s\n", i,
        br_cifara[i], br_cifara[i]==1?"": "a");
}
```