

PROGRAMIRANJE 1

PROGRAMSKI JEZIK

C

Staša Vujičić
Čas 11

NIZOVI — OSNOVNI POJMOVI

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

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]);
```

PRIMER 3

- INICIJALIZACIJA NIZA -

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

```
main()
```

```
{
```

```
/* Niz inicijalizujemo tako sto mu navodimo vrednosti  
u viticasnim zagradama. Dimenzija niza se odredjuje  
na osnovu broja inicijalizatora */
```

```
int a[] = {1, 2, 3, 4, 5, 6};
```

```
/* Isto vazi i za niske karaktera */
```

```
char s[] = {'a', 'b', 'c'};
```

```
/* Ekvivalentno prethodnom bi bilo
```

```
char s[] = {97, 98, 99}; */
```

- ⦿ Operator `sizeof()` je unarni operator koji se može primenjivati na podatke i tipove.
- ⦿ Ako se primeni na tip T, tada je vrednost izraza `sizeof(T)` broj bajtova koji zauzima jedan podatak tog tipa.
- ⦿ Ako se primeni na podatak (ili proizvoljan izraz), tada `sizeof(izraz)` ima vrednost broja bajtova koje zauzima podatak, imajući u vidu tip podatka (izraza).
- ⦿ Operator `sizeof` daje vrednost tipa `int`.
- ⦿ Vrednost izraza `sizeof` se računa u vreme prevođenja programa, jer je već tada jasno koja je veličina objekta na koji se primenjuje.

```
/* Broj elemenata niza */
```

```
int a_br_elem = sizeof(a)/sizeof(int);
```

```
int s_br_elem = sizeof(s)/sizeof(char);
```

```
/* Ispisujemo nizove */  
int i;  
for (i = 0; i < a_br_elem; i++)  
    printf("a[%d]=%d\n",i, a[i]);  
for (i = 0; i < s_br_elem; i++)  
    printf("s[%d]=%c\n",i, s[i]);  
}
```

PRIMER

- Program računa skalarni proizvod dva vektora, čije se celobrojne koordinate unose sa ulaza. Program demonstrira upotrebu nizova.

- `#include <stdio.h>`

```
int main()
{
    int a[100], b[100];
    int i, n, s;
```



```
printf("Unesite dimenziju vektora: ");
scanf("%d", &n);

printf("Uneti koordinate prvog vektora:\n");
for(i = 0; i < n ; i++)
    scanf("%d", &a[i]);

printf("Uneti koordinate drugog vektora:\n");
for(i = 0; i < n ; i++)
    scanf("%d", &b[i]);

for(i = 0, s = 0; i < n ; i++)
    s += a[i] * b[i];

printf("Skalarni proizvod vektora: %d\n", s);
}
```

PRIMER

- OBRTANJE NIZOVA -

- Program obrće elemente niza. Program demonstrira upotrebu nizova, uz korišćenje indeksne (nizovske) sintakse.

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

```
#define MAX 100
```

```
int main()  
{  
    int n;  
    int a[MAX];  
    int i,j;
```

```
/* Unosimo broj elemenata */  
printf("Uneti broj elemenata niza (<= 100): ");  
scanf("%d", &n);
```

```
/* Proveravamo da li je prekoraceno  
ogranicenje */  
if(n > MAX)  
    n = MAX;
```

```
/* Unosimo elemente niza */  
printf("Uneti elemente niza:\n");  
for(i = 0 ; i < n ; i++)  
    scanf("%d", &a[i]);
```

```
/* Prikaz niza */  
    printf("Uneli ste niz:\n");  
for(i = 0 ; i < n ; i++)  
    printf("%d ", a[i]);  
  
printf("\n");
```

```
/* Obrcemo niz */  
for(i = 0, j = n - 1 ; i < j ; i++, j--)  
{  
    int t = a[i];  
    a[i] = a[j];  
    a[j] = t;  
}
```

```
/* Prikaz niza */  
printf("Niz nakon obrtanja:\n");  
for(i = 0 ; i < n ; i++)  
    printf("%d ", a[i]);  
  
printf("\n");  
  
}
```

PRIMER

- Program za uneti datum utvrđuje redni broj tog dana u datoj godini. Program demonstrira upotrebu nizova, kao i veze nizova i pokazivača.

- `#include <stdio.h>`

```
int main()
```

```
{
```

```
int obicna[12] =
```

```
{31,28,31,30,31,30,31,31,30,31,30,31};
```

```
int prestupna[] =
```

```
{31,29,31,30,31,30,31,31,30,31,30,31};
```

```
int dan, mesec, godina;
```

```
int i, *tekuca;
```

```
int dan_u_godini = 0;
```

```
printf("Uneti datum u formatu dd:mm:yyyy:  
");
```

```
scanf("%d:%d:%d", &dan, &mesec, &godina);
```

```
/* Proveravamo da li je godina prestupna */
```

```
if(godina % 400 == 0 || (godina % 100 != 0 &&  
godina % 4 == 0))
```

```
    tekuca = prestupna;
```

```
else
```

```
    tekuca = obicna;
```

```
/* Sumiramo dane protekle prethodnih meseci.  
Indeks ide od 0 do mesec - 2, zato sto indeksi u  
C-u pocinju od 0. Npr, za dan u mesecu martu  
(treci mesec) treba sabrati broj dana u januaru i  
februaru (tj. indeks u petlji treba ici od 0 do 1,  
ukljucujuci i jednicu). */
```

```
for(i = 0 ; i < mesec - 1 ; i++)  
    dan_u_godini += tekuca[i];
```

```
/* Nakon toga jos treba dodati broj proteklih dana  
u tekucem mesecu */
```

```
dan_u_godini += dan;
```

```
printf("Uneti datum je %d. dan u godini\n",  
dan_u_godini);
```

```
}
```


PRENOS NIZA U FUNKCIJU

- Nizovi se prenose u funkciju tako što se prenese adresa njihovog početka. Iz tog razloga oni se **MOGU MENJATI** u okviru funkcije.
- Sve nizove osim niski karaktera (stringova) neophodno je prenositi zajedno sa dimenzijom niza.

PRIMER

- ◉ *Funkcija za ispis niza brojeva - demonstracija prenosa niza brojeva u funkciju.*

```
#include <stdio.h>
void print_array(int a[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
        putchar('\n');
}
```

```
main()
{
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    print_array(a, sizeof(a)/sizeof(int));
}
```

⦿ Izlaz:

⦿ 1 2 3 4 5 6 7 8 9

PRIMER

- ◉ *Skalarni proizvod dva niza brojeva.*

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

```
long mnozi(int x[],int y[],int n);
```

```
main()
```

```
{  
    int a[]={1,2,3,4,5,6}, b[]={8,7,6,5,4,3};  
    printf("Skalarno a*b= %ld\n",mnozi(a,b,6));  
}
```

```
long mnozi(int x[ ],int y[ ],int n)
{
    int i;
    long suma=0;
    for(i=0;i<n;i++)
        suma=suma+x[i]*y[i];
    return suma;
}
```

- ⊙ Izlaz:
- ⊙ Skalarno $a*b= 98$

PRIMER

- Program unosi realne brojeve sa ulaza, i računa zbir i proizvod elemenata, kao i najveći i najmanji element u nizu.

- `#include <stdio.h>`

```
#define MAX 100
```

```
/* Funkcija racuna zbir elemenata niza */  
double zbir(double a[], int n)  
{  
    double s = 0;  
    int i;  
  
    /* Sumiramo n elemenata niza. */  
    for(i = 0; i < n ; i++)  
        s=s+a[i];  
  
    return s;  
}
```



```
/* Funkcija racuna proizvod elemenata niza */  
double proizvod(double a[], int n)  
{  
    double p = 1;  
  
    /* Mnozimo n elemenata niza */  
    for(; n ; n--) p *= *a++;  
  
    return p;  
}
```

```
/* Funkcija racuna najveći element niza */
double max(double *a, int n)
{
    double max = *a;

    /* Odredjujemo najveći element */
    for(a++, n--; n > 0 ; a++, n--)
        if(*a > max)
            max = *a;

    return max;
}
```

```
/* Funkcija racuna najmanji element niza */  
double min(double *a, int n)  
{  
    double min = a[0];  
    int i;  
  
    /* Odredjujemo najmanji element */  
    for(i = 1 ; i < n ; i++)  
        if(a[i] < min)  
            min = a[i];  
  
    return min;  
}
```

```
int main()
{
    double a[MAX];
    int n , i;

    /* Ucitavamo dimenziju niza */
    printf("Uneti broj elemenata niza (<= 100): ");
    scanf("%d", &n);

    /* Proveravamo da li je dimenzija veca od MAX */
    if(n > MAX)
        n = MAX;

    /* Ucitavamo elemente niza */
    for(i = 0; i < n ; i++)
        scanf("%lf", a + i);
}
```

```
/* Pozivamo gornje funkcije i prikazujemo  
vrednosti */  
printf("Zbir elemenata niza je: %5.3f\n",  
zbir(a, n));  
printf("Proizvod elemenata niza je: %5.3f\n",  
proizvod(a, n));  
printf("Najveci element u nizu je: %5.3f\n",  
max(a, n));  
printf("Najmanji element u nizu je: %5.3f\n",  
min(a, n));  
  
}
```

PRIMER

- Program zamenjuje susedne elemente niza na parnim i neparnim pozicijama. Program demonstrira prenos nizova po adresi.

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

```
/* Deklaracije funkcija */  
void zameni(int a[], int n);  
void prikazi(int a[], int n);
```

```
int main()  
{  
    int a[] = {0,1,2,3,4,5,6,7,8,9,10};  
    int n = sizeof(a)/sizeof(int);
```

```
prikazi(a, n);
zameni(a, n);
prikazi(a, n);

return 0;
}
/* Funkcija zamenjuje susedne elemente niza na
   parnim i neparnim pozicijama. */
void zameni(int a[], int n)
{
    int i;
    for(i = 0 ; i < n - 1 ; i += 2)
    {
        int t = a[i];
        a[i] = a[i + 1];
        a[i + 1] = t;
    }
}
```

```
/* Funkcija prikazuje niz */  
void prikazi(int a[], int n)  
{  
    int i;  
  
    for(i = 0 ; i < n ; i++)  
        printf("%d ", a[i]);  
  
    printf("\n");  
}
```


PRIMER

- Program generiše niz Fibonačijevih brojeva.
- Program demonstrira vraćanje podataka nizovskog tipa iz funkcije.

- `#include <stdio.h>`

```
#define MAX 46
```

```
/* Funkcija izracunava prvih n Fibonacijevih
   brojeva i smesta ih u niz a */
void fibonaci(int a[], int n)
{
    int i;

    for(i = 0 ; i < n ; i++)
        if(i < 2)
            a[i] = 1;
        else
            a[i] = a[i - 1] + a[i - 2];
}
```

```
/* Funkcija prikazuje niz */  
void prikazi(int a[], int n)  
{  
    int i;  
  
    for(i = 0 ; i < n ; i++)  
        printf("%d ", a[i]);  
  
    printf("\n");  
}
```

```
int main()
{
    int f[MAX];
    int n;

    /* Ucitavamo dimenziju niza */
    printf("Uneti broj Fibonacijevih brojeva (<=46): ");
    scanf("%d", &n);

    /* Proveravamo prekoracenje */
    if(n > MAX)
        n = MAX;

    /* Pozivamo funkciju fibonaci() koja generise niz
       Fibonacijevih brojeva, i smesta ih u niz f. */
    fibonaci(f, n);

    /* Prikazujemo niz */
    prikazi(f, n);
}
```