

Programiranje i programski jezici



Nizovi

- Nizovi podataka predstavljaju najjednostavnije i najčešće korišćenje složene tipove. Oni služe za predstavljanje, iz matematike dobro poznatih vektora, matrica i višedimenzionalnih nizova.
- Podatak nizovnog tipa predstavlja niz podataka međusobno jednakih tipova. Oni se nazivaju elementima niza i identifikuju se pomoću rednog broja unutar niza.
- Ti redni brojevi se nazivaju indeksima elemenata niza.
- Prvi element u jeziku C obavezno ima indeks 0, drugi 1 itd.

Definisanje i inicijalizacija nizova

identifikator_niza [dužina] [dužina] ... [dužina]

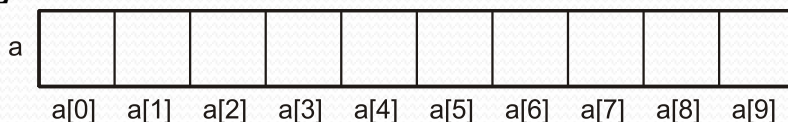
- Dužina niza predstavlja broj elemenata niza i mora biti konstantan izraz. U slučaju višedimenzionalnih nizova, dužine za sve dimenzije moraju da se pišu unutar odvojenih parova srednjih zagrada.

- Deklaracija

```
int a[10];
```

```
float a[10], mat[100][3];
```

definiše niz a veličine 10, odnosno niz od 10 uzastopnih celih brojeva $a[0]$, $a[1]$, ..., $a[9]$.



Shematski prikaz niza a u memoriji računara.

- Notacija $a[i]$ ukazuje na i -ti element niza.

Primer ilustruje inicijalizaciju nizova.

```
#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}; */
/* 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]);
}
```

Zadatak 1

Napisati program koji učitava sa standardnog ulaza broj n ($n \leq 100$), a potom n elemenata niza celih brojeva. Program treba da ispiše maksimum unetog niza na standardni izlaz.

Zadatak 1

```
#include <stdio.h>
#define DIM 100
int max( int prvi, int drugi );
main()
{
    int a[DIM], indeks, n, Max;
    printf("\nUnesite dimenziju niza: "); scanf("%d", &n);
    printf("\nUnesite clanove niza: ");
    for(indeks=0; indeks<n; indeks++) scanf("%d", &a[indeks]);

    Max=a[0];
    for(indeks=1; indeks<n; indeks++) Max=max(Max, a[indeks]);
    printf("\nMaksimum niza je %d \n", Max);
}
/* max: vraca maksimum dva argumenta */
int max( int prvi, int drugi )
{
    return ( (prvi>=drugi)?prvi:drugi );
}
```

Pokazivači i nizovi

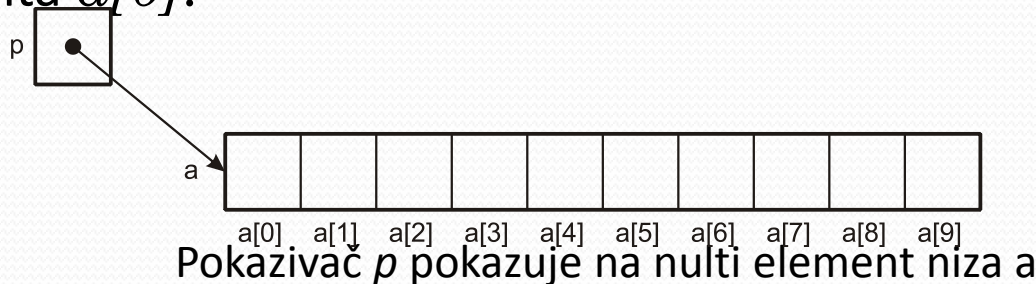
- U C-u postoji jaka veza između pokazivača i nizova
- Ukoliko je p pokazivač na ceo broj, deklarisan kao

```
int *p;
```

onda linija

```
p = &a[0];
```

postavlja pokazivač p da pokazuje na nulti element niza a (indeksi nizova u C-u počinju od nule), ili drugačije rečeno pokazivač p sadrži adresu elementa $a[0]$.



- Sada linija

$x = *p;$

dodeljuje promenljivoj x vrednost elementa $a[0]$.

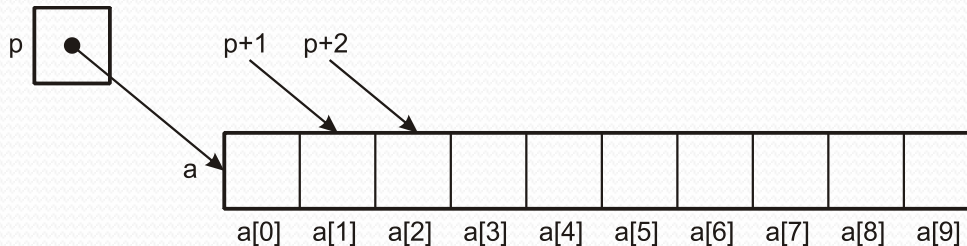
- Ukoliko pokazivač p pokazuje na određeni element niza, po definiciji $p+1$ pokazuje na sledeći element, $p+i$ pokazuje na i -ti element iza p , a $p-i$ pokazuje na i -ti element ispred p . Tako, ukoliko p pokazuje na $a[0]$, onda

$*(p+1)$

predstavlja sadržaj elementa $a[1]$, a

$*(p+i)$

sadržaj elementa $a[i]$.



Pristupanje elementima niza pomoću pokazivača

- Indeksiranje nizova i aritmetika pointera su veoma bliski.
- Po definiciji, naziv niza je upravo adresa nultog elementa niza. Zbog toga nakon izvršenja linije

```
p = &a[0];
```

p i a imaju jednake vrednosti, tako da prethodna linija može biti napisana i kao

```
p = a;
```

- $a[i]$ se može napisati kao $*(a+i)$
- Ukoliko primenimo operator $\&$ na oba ova oblika, dobijamo da je $\&a[i]$ isto što i $a+i$, tj. adresa i -tog elementa niza.
- Slično važi i za pokazivač p , pa je $p[i]$ isto što i $*(p+i)$.
- Postoji razlika između pokazivača i nizova

$p=a$	i	$p++$	je dozvoljeno
$a=p$	i	$a++$	nije dozvoljeno

- Kada se funkciji prosleđuje naziv niza, ono što se zapravo šalje je adresa nultog elementa.
- Unutar pozvane funkcije ovaj argument je lokalna promenljiva, a parametar koji predstavlja niz je pokazivač, odnosno promenljiva koja sadrži adresu nultog elementa niza.
- Posmatrajmo funkciju za određivanje dužine stringa (niza karaktera):

```
/* duzina: vraca duzinu stringa s */  
int duzina(char *s)  
{  
    int n;  
  
    for(n = 0; *s != '\0'; s++) n++;  
  
    return n;  
}
```

...

```
duzina("Zdravo, drugari!");    /* konstantan string */  
duzina(niz);    /* char niz[100]; */  
duzina(pok);    /* char *pok */
```

- U definiciji funkcija dozvoljeno je deklarisanje formalnih parametara na oba načina; i kao niza i kao pokazivača.

- Deklaracije

```
char s[]
```

```
i
```

```
char *s
```

su ekvivalentne.

- Funkciji je moguće proslediti i deo nekog niza tako što bi joj se prosledio pokazivač na podniz. Tako, ukoliko je a niz, pozivi

```
fun(&a[5]);
```

```
i
```

```
fun(a+5);
```

prosleđuju funkciji *fun* podniz koji počinje od petog elementa niza a .

Zadatak 2

Napisati funkciju za ispis niza brojeva - demonstrira prenos nizova brojeva u funkciju.

```
#include <stdio.h>
void print_array( int a[], int n) {
    int i;
    for (i = 0; i < n; i++) { a[i]++; printf("%d ",a[i]);}
    putchar('\n');
    /* Obratite paznju na ovaj sadrzaj koji se stampa!!! */
    printf("sizeof(a) - u okviru fje : %ld\n", sizeof(a));
}
main() {
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    int i;
    printf("sizeof(a) - u okviru main : %ld\n", sizeof(a));
    int n= sizeof(a)/sizeof(int);
    print_array(a, n);
    for (i = 0; i < n; i++) printf("%d ",a[i]);
    putchar('\n');
}
```



Zadatak 3

Napisati program koji će ispisati rezultat skalarnog množenja uređene šestorke a i uređene šestorke b. Uređenu šestorku a čini prvih šest prirodnih brojeva, a uređenu šestorku b čine brojevi 8,7,6,5,4,3

Zadatak 3

```
#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("Skalaro a*b= %ld\n",mnozi(a,b,6));
}

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

Zadatak 4

Napisati program koji prihvata sa standardnog ulaza pozitivan ceo broj n ($n \leq 50$), a zatim prihvata po jedan element n -dimenzionalnog niza celih brojeva. Formirati drugi niz koji sadrži samo nenegativne elemente unetog niza, a potom članove drugog niza ispisati na standardni izlaz.

Zadatak 4

```
#include <stdio.h>
main() {
    int n, indeks, a[50], j, rezultat[50];

    do {
        printf("Unesite broj elemenata niza\n");
        scanf("%d", &n);
    } while (n < 1 || n > 50);

    for( indeks=0; indeks<n; indeks++) scanf("%d", &a[indeks]);

    for( indeks=0, j=0; indeks<n; indeks++) {
        if(a[indeks]<0) continue;
        rezultat[j++]=a[indeks];
    }
    printf("\nNovi niz je: ");
    for( indeks=0; indeks<j; indeks++) printf("%d\t", rezultat[indeks]);
    printf("\n");
}
```