

Nizovi

- ❖ Treće pravilo: ugrađeni operator indeksiranja $x[y]$, gde su x i y izrazi, izračunava se uvek kao $*(x+y)$

- ❖ Na primer:

```
int a[10];  
int* p = a;
```

p ukazuje na $a[0]$

```
a[2]=1;
```

$a[2]$ se izračunava kao $*(a+2)$; pošto je a niz, konvertuje se u pokazivač na svoj prvi element; zbir tog pokazivača i celog broja 2 je vrednost koja ukazuje na $a[2]$

```
p[3]=3;
```

$p[3]$ se izračunava kao $*(p+3)$; pošto je p ukazuje na $a[0]$, zbir tog pokazivača i celog broja 3 je vrednost koja ukazuje na $a[3]$

```
p=p+1;
```

p sada ukazuje na $a[1]$

```
*(p+2)=1;
```

$p+2$ ukazuje na $a[3]$ koji dobija vrednost 1

```
p[-1]=0;
```

Isto što i $*(p+(-1)) = 0$

```
2[p-1] = 0;
```

Isto što i $*(2+(p-1)) = 0$

Nizovi

- ❖ *String literali (string literals)* predstavljaju literale tipa “niz (konstantnih) znakova”
- ❖ Navode se između navodnika, uz mogućnost definisanja tipa znakova i načina kodovanja slično kao i za znakovne literale; između navodnika mogu da se nađu odgovarajući znakovi i *escape* sekvence, kao za znakovne literale; na primer:

```
"I said: \"Hello!\""
```

```
L"Hello"
```

```
u8"Hello"
```

```
u"Hello"
```

```
U"Hello"
```

- ❖ Dva susedna string literala, razdvojena samo belinama (uključujući i znak za novi red) spajaju se u jedan nakon pretprocesiranja; ovako se dugački string literali mogu nastavljati u novi red:

```
"This is a very, very long string literal that cannot fit in "  
"one single line of code, but has to be split in two lines."
```

- ❖ Isto se može koristiti za specijalne potrebe, kada bi se dati niz susednih znakova između navodnika tumačio kao *escape* sekvenca:

```
"\0" "1"
```