

Nizovi

- ❖ Niz može da se inicijalizuje tzv. *agregatnim inicijalizatorom* (*aggregate initializer*) između velikih zagrada; tada ne mora da se navede dimenzija niza, ona se određuje na osnovu broja elemenata u inicijalizatoru i smatra se definisanom:

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

b je niz od 3 elementa tipa *int*, tj. tipa *int[3]*

```
int b[] = {4, 5, 6};
```

```
int c[3] = {1};
```

c je niz od 3 elementa tipa *int*, inicijalizovanih redom na 1, 0 i 0

- ❖ Niz može da se deklarise i bez zadavanja (prve) dimenzije (ako je u pitanju niz nizova, sve ostale dimenzije moraju da se zadaju, jer one definišu tip elementa niza); ovaj niz je nekompletnog tipa (osim ako je inicijalizovan agregatnim inicijalizatorom). Za ovakve nizove mogu se praviti pokazivači i reference na njih, ali se ti pokazivači i reference ne mogu inicijalizovati tako da ukazuju na nizove sa poznatim dimenzijama. Takvi pokazivači se mogu dereferencirati i prenositi kao parametri funkcija, ali ne mogu učestvovati u pokazivačkoj aritmetici:

```
extern int a[];
```

p je pokazivač na niz nekompletnog tipa (nepoznate veličine)

```
int (*p)[] = &a;
```

Greška u prevođenju (ovo je ispravno na jeziku C)

```
int (*q)[2] = &a;
```

- ❖ Nizovi ne mogu da se kopiraju operacijom dodele, tj. ne mogu biti levi operandi operatora dodele. Međutim, ako je niz podatak član nekog klasnog tipa (strukture ili klase), on se može kopirati kao deo objekta kome pripada:

```
int a[10], b[10];
```

```
class X { public: int a[10]; };
```

```
X x1, x2;
```

Greška u prevođenju

```
a = b;
```

```
x1 = x2;
```

Niz *x2.a* kopira se u niz *x1.a*

Nizovi

- ❖ Zbog svoje izvorne orijentacije na efikasnost generisanog koda, operacije sa nizovima na jezicima C i C++ rade se sa najmanje moguće potrebnih informacija; da bi se izvršila operacija pristupa elementu niza, potrebno je, pored izračunavanja izraza koji određuje vrednost indeksa, znati samo sledeće:
 - početnu adresu (prvog elementa) niza i
 - veličinu (svakog) elementa niza, koja je poznata za vreme prevođenja iz tipa elementa tog niza

Pomoću ovih informacija, generisani kod za pristup elementu niza tipa T izračunava adresu indeksiranog elementa na sledeći način:

$$adresa_elementa := adresa_početka_niza + vrednost_indeksa * sizeof(T)$$

- ❖ Informacija o adresi i tipu elementa sadržana je i u pokazivaču tipa T^*
- ❖ U saglasnosti sa ovim, nizovi se ne prenose kao parametri funkcija, niti se mogu vraćati kao vrednosti funkcija (kao kompletni paketi elemenata); umesto toga, prenose se i vraćaju pokazivači na početak (prvi element) niza (ili reference na nizove, potpuno slično)
- ❖ Prema tome, pozvana funkcija nema implicitnu informaciju o stvarnoj veličini niza, nego se ona mora preneti funkciji na neki drugi način: ili preko posebnog parametra funkcije, ili niz mora imati neku posebnu vrednost koja označava njegov poslednji element (terminiran niz)
- ❖ Zbog svega ovoga, nizovi i pokazivači povezani su sledećim pravilima jezika