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

Nizovi

- Prvo pravilo: postoji implicitna konverzija niza elemenata tipa *T* u pokazivač na tip *T*, koja se vrši na svakom mestu gde se očekuje pokazivač, a pojavljuje se niz; vrednost ovog pokazivača ukazuje na prvi element datog niza (tzv. "rastakanje" niza u pokazivač, *array-to-pointer decay*):

  T[] ⟹ *T**

  Na primer:

```
void f (int a[]) { cout<<a[0]<<endl; }

void g (int* p) { cout<<*p<<endl; }

int main () {

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

  f(a);
  f(p);

  g(a);
  g(p);

}
```