

Znaki

Znak (`char`) je poseben celoštevilski tip. V spremenljivkah tega tipa hranimo kode znakov. Ker si je kode posameznih znakov težko zapomniti, lahko namesto njih v programu pišemo kar pripadajoče znake, obdane z enojnimi narekovaji, npr. namesto vrednosti 65 lahko napišemo tudi `'A'`.

```
char znak = 'A';
int koda = (int)znak;
```

Z znaki lahko računamo kot s celimi števili, lahko jih primerjamo z drugimi znaki, ter seveda beremo in izpisujemo. Formatno določilo za branje in izpis znaka je `%c`. Nekaj težav lahko nastopi pri branju znaka, saj bo program namesto vpisanega prebral morebiten znak za novo vrstico iz prejšnjega vnosa. Temu se izognemo s klicem funkcije `fflush`, ki ji za parameter damo ime standardnega vhoda (`stdin`).

```
fflush(stdin);
scanf("%c", &znak);
printf("%c", znak);
```

Kod znakov se nima smisla učiti na pamet. Dovolj je, da vemo, da imajo številke zaporedne kode (`'0'` ima kodo 48, `'1'` ima kodo 49, ... `'9'` ima kodo 57). Podobno velja za velike črke angleške abecede (`'A'` ima kodo 65, ...) in male črke angleške abecede (`'a'` ima kodo 97, ...).

Če imamo znak, ki predstavlja neko številko, lahko tako zelo preprosto izračunamo njegovo vrednost (iz znaka `'5'` bi radi dobili vrednost 5). Od znaka je dovolj samo odšteti znak `'0'`. Če pa imamo vrednost med 0 in 9 in bi želeli dobiti ustrezen znak, moramo znak `'0'` prišteti in dobljeno vrednost pretvoriti v znak.

```
int vrednost = znak - '0';
char znak = (char)(vrednost + '0');
```

Da imajo omenjene skupine znakov zaporedne kode, lahko izkoristimo tudi pri ugotavljanju, kakšne vrste je dani znak. Če nas zanima, ali je znak mala črka, bi sestavili takšen pogoj:

```
if (znak >= 'a' && znak <= 'z') ...;
```

Omeniti je potrebno še nekaj posebnih znakov:

- `'\"'` dvojni narekovaj
- `'\''` enojni narekovaj
- `'\\'` nagibnica
- `'\n'` skok v novo vrsto
- `'\t'` tabulator
- `'\0'` znak s kodo 0