

Vstup a výstup

Datové typy

C je striktně typovaný jazyk. Mnoho datových typů si můžeme definovat sami, pro začátek si vyzkoušíme následující:

- `char` - zabírá prakticky vždy 1B, reprezentuje znak, tj. 65 má význam `A`, 66 `B`, 30 `0` atd. Toto číslování platí pro dolních 7 bitů, 8. bit se může¹ využívat jako znaménkový, typ má pak rozsah -128 až +127. Bezznaménkový je 0 až 255. Chceme-li vynutit znaménkový, použijeme `signed char`, pro bezznaménkový `unsigned char`. Znakovou konstantu zapisujeme do apostrofů (např. `'A'`).
- `int` - je číslo, které zabírá alespoň 2 byty², čímž je omezen rozsah. Je vždy znaménkové, nepoužijeme-li `unsigned`. Číslo je optimalizováno pro rychlé operace na dané architektuře, chceme-li použít přesný rozsah, v souboru `stdint.h` jsou definovány typy jako `uint32_t` (bezznaménkové 32bitové číslo), `int16_t` (znaménkové 2bytové) a další.
- `float` a `double` je znaménkové desetinné číslo s pohyblivou (plovoucí, *floating*) desetinnou čárkou (tečkou) dle standardu IEEE 754. Nelze deklarovat jako bezznaménkové. `float` zabírá 4 byty, `double` má dvojnásobnou přesnost a zabírá 8 bytů. Později si popíšeme podrobněji.
- dále je možné do proměnných uložit i text - to si vysvětlíme později, prozatím budeme pracovat jen s textovými konstantami zapsanými pomocí uvozovek.

Deklarace a definice

Proměnná je platná v bloku (ohrazeném `{ }`), kde je deklarována. Deklarace se provádí uvedením datového typu a pouze jednou. Definice je prvotní přiřazení hodnoty proměnné:

```
int x; // deklarace
x = 42; // definice
int y = 24; // deklarace spojená s definicí
{
    int y = 33; // tato proměnná platí jen uvnitř bloku
    // zde x=42, y=33
}
// zde x=42, y=24
```

Formátovaný výstup

K formátovanému výstupu slouží funkce `printf`, která má proměnný počet parametrů. První je vždy text, který může obsahovat zástupné symboly (*tokens*) `%`, které jsou nahrazeny postupně dalšími parametry s následujícím významem:

¹ standard toto neomezuje, některé kompilátory používají bezznaménkovou variantu, jiné znaménkovou

² na AVR mikrokontrolerech (na Arduino) má 2 byty, na 32bitových Windows na PC 4 byty (32 bitů), na Linuxech na PC 8 bytů

- %% je nahrazeno znakem %
- %d je nahrazeno číslem (int)
- %f je nahrazeno desetinným číslem (float či double)
- %.2f je nahrazeno desetinným číslem, zobrazí se 2 desetinná místa
- %c je nahrazeno znakem (char)

Co se znaků týče, speciální význam má \, která zobrazuje následující znak s alternativním významem; nejčastěji se používá \n, což je znak s ordinální hodnotou 13 (konec řádky)

```
char a = 'A';
printf("znak %c\n", a); // znak A
printf("%c=%d\n", a, a); // A=65
```

Formátovaný vstup

K formátovanému vstupu slouží funkce `scanf`, která má stejný význam parametrů, ale proměnné načítá. Jako parametry funkcí je třeba zadat jejich adresy. Adresa proměnných typů, které jsme si uvedli, se získá operátorem `&`:

```
int x;
scanf("%d", &x);
printf("%d", x); // vytiskne to, co uživatel zadal
int y;
scanf("%d,%d", &x, &y); // načte dvě čísla oddělená čárkou
```

Funkce `scanf` čte parametry až do stisknutí enter, znak konce řádky (`\n`) ale ponechá ve vstupní paměti, stejně jako všechny znaky, které tam zůstaly. Ty lze vyjmout opakovaným voláním `getchar`:

```
int n;
scanf("%d", &n); // zadejte např. 12 jablek
while(getchar() != '\n'); // vyhození " jablek" ze vstupní fronty
printf("vstup: %d\n", n); // zobrazí 12
getchar(); // čeká na stisk klávesy
```