Datové typy:  
Název Velikost Popis  
char 1 byte znak  
int/long 4 bytes Celé číslo: tj. 97 nebo hex 0x61, oct 0x141  
long long 8 bytes Velmi velké celé číslo  
float 4 bytes desetinná plovoucí hodnota  
double 8 bytes dvojnásobně dlouhý float  
char, int, a double jsou nejpoužívanější a jednoduše používaný v malých programech. Sizeof (double) vypočítá velikost dvojnásobku v adresovatelných jednotkách (bytes). Nulové hodnoty představují logické nepravdivé, nenulové hodnoty jsou logické pravdivé.

cyklus while - while (podmínka)  
 {příkazy}  
cyklus for – for (int i = 0; i < N; i++)  
 {příkazy}  
cyklus do {   
do-while příkazy  
 } while (podmínka);  
příkaz switch - switch (výraz)   
(V příkazu case {   
nejde použít case konst1: příkazy break;  
konstanty case konst2: příkazy break;   
definované default: příkazy   
pomocí const.) }

konec příkazů - switch, while, do, for, break;  
skok na další iteraci - while, do, for, continue;  
ukončení a návrat hodnoty funkce - return výraz;  
 příkaz if - if (podmínka)  
 {příkazy}   
 else if (podmínka)  
 příkaz;  
 else  
 {příkazy}   
Pokud je před else blokový příkaz, středník se nepíše. Pokud je tam samostatný příkaz, musí být ukončen středníkem.

Pole:  
jednorozměrné pole |typ jmeno [N];  
vícerozměrné pole |typ jmeno [K] [L] [M]...;  
statická alokace |int pole [ROZMER];  
 |int matice[RADKU][SLOUPCU];  
+ inicializace |int pole [] = {1, 2, 3};  
++ všech prvků nulou|int pole[ROZMER] = {0};  
→ {1, 0, 0, 5, 0} |int pole [5] = {1, [3]=5};  
Nevyjmenované prvky inicializovaného pole mají hodnotu nula.

Komentáře:  
 znaky zprava od // je jako komentář. Text mezi /\* a \*/ může být jako víceřádkový komentář.

Znak Argument (obsahy proměnné)  
%c jeden znak  
%d Desítkové celé číslo se znaménkem. (int)  
%e Plovoucí desetinná čárka  
%f Hodnota v desetinných číslech (float)  
%g celá hodnota ve formátu %e nebo %f, podle toho, co je kratší  
%i Desítkové celé číslo se znaménkem. (int)  
%o Osmičkové celé číslo bez znaménka. (int)  
%s Určuje jeden bajt nebo vícebajtový řetězec znaků.  
%u Celé číslo bez znaménka. (int)  
%x Šestnáctkové celé číslo bez znaménka; používá. (int)  
%% (Znak procenta)

Symbol |význam | příklad  
< | méně než | 1 < 5 ; 8 < 9  
== | rovno | 5 == 5 ; 0 == 0  
> | větší než | 8 > 5 ; 10 > 0  
<= | méně nebo rovno | 4 <= 5 ; 8 <= 8  
>= | větší nebo rovno | 9 >= 5 ; 2 >= 2  
!= | není rovno | 1 != 0 ; 4 != 3.99

Funkce C knihoven:

<stdio.h> standartní I/O

<conio.h> console I/O

<string.h> funkce řetězců

<stdlib.h> obecné funkce

<math.h> mat. funkce

<time.h> funkce času

<ctype.h> zpracování znaků

<stdarg.h> fce prom. argu.

<signal.h> funkce signálu

<setjmp.h> funkce skoku

<locale.h> lokální funkce

<errno.h> zpracování chyb

<assert.h> fce diagnostiky

Logická hodnota – bool. Hodnoty: true, false.  
prázdný typ – void  
obecný typ (bez hodnoty) – void  
obecný ukazatel – typ\*  
ukazatel na číslo – int \*, float \*, …  
textový řetězec (končí nulou) – char jmeno [N] (kompatibilní s (char \*))  
struktura – struct jmeno {…};  
výčet – enum jmeno {…};  
přejmenování typu – typedef typ noveJmeno;  
zjištění velikosti – sizeof (výraz); sizeof (typ);  
přetypování výrazu – (typ) výraz

Celé číslo v soustavě:  
 – desítkově 12345   
– osmičkové 012345 (prefix 0 – nula)   
– hexadecimální 0x123AB (prefix 0x nebo 0X) číslo typu long - 1234L (přípona L)   
číslo typu double - 123.0   
číslo v exp. Tvaru - 1.23e12   
znak - ’a’  
znak (osmičkově, hexa) - ’\101’, ’\x41’   
nový řádek, cr, tabulátor ’\n’ ’\r’ ’\t’

speciální znaky - ’\\’ ’\?’ ’\’’ ’\”’  
 řetězec (končí nulou ’\0’) - char jm[] = ”hubero”; // sizeof(jm) == 7   
řetězec (znaky na indexu 6–9 jsou ’\0’) - char jm[10] = ”kororo”; // sizeof(jm) == 10   
POZOR! strlen(jm) == 7, ale sizeof(jm) == 4 - char \*jm = ”ororok”; // sizeof(jm) != 7

|  |  |  |  |
| --- | --- | --- | --- |
| Pr. | Význam | Operátory | As. |
| 1. | Volání funkce  indexace pole  přístup ke složkám struktury | ()  []  Jmeno.člen  Ukazatel - > člen | → |
| 2. | Inkrementace, dekrementace  Unární +, -  Logický, bitový not  Reference, deference  Přetypování  Velikost objektu | ++, --  +, -  !, -  & jmeno, \*ukazatel  (typ) proměnná  Sizeof | ← |
| 3. | Násobení, dělení, modulo | \*, /, % | → |
| 4. | Sčítání, odčítání | +, - | → |
| 5. | Bitový posun vlevo, vpravo | <<, >> | → |
| 6. | porovnávání | >, >=, <=, < | → |
| 7. | porovnávání | ==, != | → |
| 8. | Bitový and | & | → |
| 9. | Bitový xor | ˆ | → |
| 10. | Bitový or | | | → |
| 11. | Logický and | && | → |
| 12. | Logický or | || | → |
| 13. | Podmíněný výraz | Podmínka ? výraz1 : výraz2 | → |
| 14. | Přiřazení | =, +=, -=, \*=, /=, %=, >>=, <<=, &=, |=, ˆ= | ← |
| 15. | čárka | , | → |

Vstupy a výstupy:  
Znaky:  
 getchar() - Vrací ANSI kód jednoho znaku z bufferu vstupního streamu (proudu) jako celé číslo   
putchar(int) - Vytiskne jeden znak zadaného ANSI kódem celým číslem do bufferu výstupního streamu (proudu) Řetězce:   
gets(strJmeno) - Přečte řádku ze vstupního streamu (proudu) do proměnné typu string (řetězec) fgets(strJmeno, delka, stdin); - Přečte řádku ze vstupního streamu (proudu) do proměnné typu string (řetězec) puts("string") - Vytiskne řetězec do výstupního streamu - proudu

Formátovaná data:  
scanf("%d", &x) - Načte hodnotu/hodnoty (typ je definován formátovacím řetězcem) do  
 proměnné/proměnných (typ se musí  
 shodovat) ze vstupního streamu.  
 Přestane číst v okamžiku prvního výskytu  
 prázdného znaku. & prefix není zapotřebí  
 pro pole (včetně řetězců) (není jisté použití)   
printf("I love %c %d!", 'C', 99) - Tiskne data (ve formátu  
 definovaném formátovacím řetězcem)  
 jako řetězec do výstupního streamu.   
fgets(strName, length, stdin); sscanf(strName, "%d", &x); Použijte funkci fgets pro omezení vstupní délky, potom použijte sscanf pro načtení výsledného řetězce na místo funkce scanf. (bezpečné použití)