

2 – Skalární datové typy

Příklad 2.1

Následující příklad ukazuje, jak zjistit délku proměnné daného typu v bytech. Délka proměnné není dána normou jazyka C, ale velikost daného typu můžeme zjistit buď v souboru limits.h nebo pomocí operátoru sizeof.

```
#include<stdio.h>
#include<conio.h>
main()
{
clrscr();
printf("cele: %d\n",sizeof(int));
printf("kratke cele: %d\n",sizeof(short int));
printf("dlouhe cele: %d\n",sizeof(long int));
printf("float: %d\n",sizeof(float));
}
```

Příklad 2.2

Zjistěte velikosti ostatních skalárních typů : unsigned short int, unsigned long int, signed char, unsigned char, double, long double, pointer.

Příklad 2.3

Tento příklad ukazuje, k jaké dojde chybě při špatně zvoleném formátu výpisu. Pro výpis celých čísel long nutno přidat ke znaku konverze ještě modifikátor l nebo L. Podrobněji skripta kap. Vstup a výstup.

```
#include<stdio.h>
#include<conio.h>
main()
{
clrscr();
long int k = 111111111;
printf("%d",k); /* nutno za menit za printf("%ld",k) */
}
```

Příklad 2.4

Napište desítkový, osmičkový a šestnáctkový ekvivalent následujících dvojkových čísel:

- 00010010
- 01100101
- 01101011
- 10111011
- 00111111
- 00001100 01100100

Při vyhodnocení předpokládejte, že počítač užívá tzv. dvojkové doplňkové notace tj.

- zcela levý bit je využit na znaménko , pokud je 1, je číslo záporné, jinak kladné.
- pro vytvoření záporného čísla převrátíme všechny bity a poté přičteme jedničku. tj. -1 je 1 (00000001), převrátíme (11111110) a přičteme jedničku (11111111- to je výsledná hodnota)

Příklad 2.5

Následující příklad ukazuje, jak C provádí konverzi při přiřazení. Typ na pravé straně je automaticky změněn na typ na straně levé. Pokud je typ na straně levé kratší, dochází k zachování nižšího počtu bytů. Vysvětlete chování následujícího příkladu.

```
#include<stdio.h>
#include<conio.h>
main()
{
int i = 882; /*00000011 01110010 */
}
```

```
char c;  
c = i;    /* 01110010 */  
printf("%d %c",i,c);  
}
```

Příklad 2.6

Napište příklad, který ukáže, jak jazyk C konvertuje long na short.

Příklad 2.7

Napište příklad, který ukáže, jak jazyk C konvertuje float na short.

Příklad 2.8

Napište příklad, který ukáže, jak jazyk C konvertuje float na char.

Příklad 2.9

Napište příklad, který ukáže, jak jazyk C konvertuje double na int (např. double je 6.88), to samé pro záporné číslo (-9.125).

Příklad 2.10

Zjistěte, jak pracuje funkce printf() při výpisu float a double. Užijte pro výpis %f. Kolik se vytiskne desetinných číslic? Dochází k zaokrouhlení nebo odseknutí?

Příklad 2.11

Co vypíše tento program, nejprve odhadněte, poté proveďte.

```
#include<stdio.h>  
main()  
{  
    int j;  
    float f =12.4589;  
    printf("tisknu float %f",f);  
    j= (int)f;  
    printf("tisknu int %d",j);  
    f= (float)j;  
    printf("tisknu float %f",f);  
}
```

Příklad 2.12

Následující funkce umí zkonvertovat teplotu ve stupních Fahrenhita do stupňů Celsia. Napište funkci opačnou: z Celsia do Fahrenhita, poté obě funkce použijte v programu.

```
/* Zkonvertuje float hodnotu z Fahrenheitů do Celsia */  
  
double fahrenheit_to_celsius ( double temp_fahrenheit )  
{  
    double temp_celsius;  
    temp_celsius = (temp_fahrenheit - 32.0) *  
        100.0/(212.0 - 32.0);  
    return temp_celsius;  
}
```

Příklad 2.13

Co vypíše následující program, proč? Program upravte pomocí přetypování, tak aby pracoval dobře.

```
#include<stdio.h>  
main()  
{  
    int j = 2, k = 3;  
    float f;
```

```
f = k / j;  
printf( " vypis f = %f",f);  
}
```

Příklad 2.14

Napište následující deklarace:

- a) celé dlouhé číslo bez znaménka
- b) reálné číslo s dvojitou přesností
- c) pointer na znak
- d) znak inicializovaný hodnotou A
- e) externí funkce vracející celé číslo bez znaménka

Příklad 2.15

Napište program, který umí spočítat cenu včetně daně. Daň je buď 22% nebo 5%. Na to, jaká je cena zboží bez daně a jaká daň se užije, se program musí zeptat uživatele. Užijte hodnoty float.

Příklad 2.16

Následující program vypíše z reálného čísla pouze celou část. Povšimněte si, která část programu pracuje dobře a poté program doplňte naopak o výpis desetinné části. Tj. je - li zadáno číslo 45.879, program vypíše 0.879.

```
#include <stdio.h>  
main()  
{  
    double g;  
    int i;  
    printf("Zadejte realne cislo s desetinnou casti : ");  
    scanf("%lf", &g);  
    printf(" tisk bez desetinne casti : %.0f - pomoci formatu tisku \n", g);  
    printf(" tisk bez desetinne casti : %d - pomoci pretypovani \n", (int) g);  
    i = g;  
    printf(" tisk bez desetinne casti : %d - pomoci implicitni typove konverze \n", i);  
}
```

Příklad 2.17

Následující program převádí velká písmena na malá. Přepište program s využitím funkcí *toupper()* a *tolower()*. Jejich popis si najděte v Helpu.

```
#include <stdio.h>  
main()  
{  
    int c1, c2, c3;  
    int vzdalenost;  
    printf("Zadejte 3 velka pismena najednou , ukoncete Enter : ");  
    vzdalenost = 'a' - 'A';  
    c1 = getchar() + vzdalenost;  
    c2 = getchar() + vzdalenost;  
    c3 = getchar() + vzdalenost;  
    printf("Mala pismena jsou : %c%c%c\n", c1, c2, c3);  
}
```

Příklad 2.18

Následující program ukazuje některé záludnosti automatické konverze v C. Nejprve odhadněte, co program vytiskne, poté porovnejte s výsledkem. Výsledek odůvodněte.

```
#include<stdio.h>  
main()  
{  
    char d = 6;  
    char c = 2;
```

```
double g = 1.2;
int i;
d = d+'4';
i = g * c;
printf(" tisknu kolik je d ,d = %c ",d);
printf(" tisknu kolik je i,i = %d",i);
}
```

Příklad 2.19

Tento program ukazuje, jaké nesmysly se dají dělat s výčtovým typem. Je to ale zcela zbytečné. Co program vytiskne?

```
#include<stdio.h>
enum ptaci{ zeleny,hnedý=12, modro_bily};
int f()
{
return (int) modro_bily;
}

main()
{
printf("co vraci funkce f %d",f());
}
```

Příklad 2.20

Program ukazuje základní práci s pointry. Jaká je hodnota j a i na konci programu?

```
#include<stdio.h>
main()
{
int i,j, *ip;
ip = &i;
i = 22;
j = *ip;
*ip = 17;
printf ("hodnota j je %d a hodnota i je %i",j,i);
}
```

Příklad 2.21

Napište program,který užije následující deklarace:

```
char z;
int j,k,l;
float s,r,t;
```

Poté program vypíše adresy těchto proměnných. Jak kompilátor používá paměť pro umístění proměnných?

Od RNDr. Dagmar Brechlerové.