

Zdrojové kódy v jazyce C

Miroslav Jílek

Všechny kódy byly kompilovány a kompilované soubory byly testovány.

Ukázka programu s použitím funkce, pointeru ve funkci, alokací paměti pro hodnotu integer a vložení adresy do pointeru.

*int *b;*

vytvoření místa v paměti, do kterého bude uložena adresa odkazující na místo v paměti, kde je uložen integer

b=(int)malloc(sizeof(int));*

vytvoření místa v paměti pro integer, adresa tohoto místa se vloží do pointeru b

#include <stdio.h>

#include <stdlib.h>

*void mojefunkce (int *a)*

{

**a=10;*

}

int main(void)

{

*int *b, c;*

b=(int)malloc(sizeof(int));*

mojefunkce(b);

*c=*b;*

printf("%d\n", c);

return 0;

}

**Ukázka programu, kde je použita struktura (struct).
Struktura je zde použita v návratové hodnotě funkce.**

```
#include <stdio.h>
#include <stdlib.h>

typedef struct BALIKDAT
{
    int Hodnota1;
    int Hodnota2;
} Balikdat;

Balikdat MojeFunkce (int Vstup)
{
    Balikdat Data;
    Data.Hodnota1=3*Vstup;
    Data.Hodnota2=3+Vstup;
    return Data;
}

int main (void)
{
    int C=10;
    Balikdat NovaData;
    NovaData=MojeFunkce(C);
    printf("Hodnota1 je %d, Hodnota2 je %d.\n", NovaData.Hodnota1,
NovaData.Hodnota2);
    return 0;
}
```

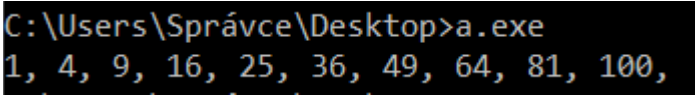
Hodnota1 je 30, Hodnota2 je 13.

Ukázka programu s rekurzivní funkcí (zobrazí mocniny čísel od 1 do 10).

```
#include <stdio.h>

void mocniny (int C)
{
    C++;
    printf("%d, ",C*C);
    if (C==10) return;
    mocniny(C);
}

int main (void)
{
    mocniny(0);
    return 0;
}
```



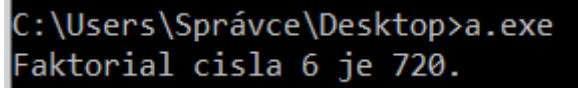
```
C:\Users\Správce\Desktop>a.exe
1, 4, 9, 16, 25, 36, 49, 64, 81, 100,
```

Ukázka rekurzivní funkce (výpočet faktoriálu).

```
#include <stdio.h>

int faktorial (int e)
{
    if (e<2) return 1;
    return e * faktorial(e-1);
}

int main (void)
{
    int a=6;
    printf("Faktorial cisla 6 je ");
    a=faktorial(a);
    printf("%d.\n",a);
    return 0;
}
```

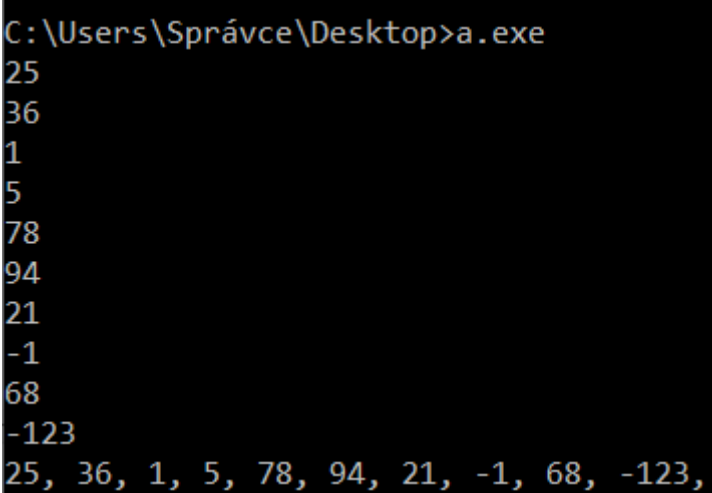


```
C:\Users\Správce\Desktop>a.exe
Faktorial cisla 6 je 720.
```

Ukázka programu – vstup deseti integerů do statického pole.

```
#include <stdio.h>

int main (void)
{
    int pole[10], i;
    for (i = 0; i < 10; i++)
    {
        if (scanf("%d", pole + i) != 1)           // pole + i <=> &(pole[i])
        {
            printf("Nespravny vstup\n");
            return 1;
        }
    }
    for (i = 0; i < 10; i++) printf("%d, ", pole[i]);
    return 0;
}
```



```
C:\Users\Správce\Desktop>a.exe
25
36
1
5
78
94
21
-1
68
-123
25, 36, 1, 5, 78, 94, 21, -1, 68, -123,
```

Ukázka programu se vstupem do dynamického pole integerů s realokací.

```
#include <stdio.h>
#include <stdlib.h>

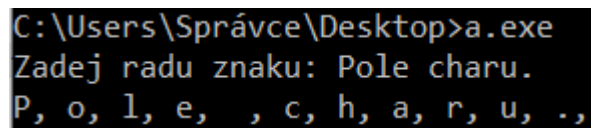
int main (void)
{
    int i=0, j=10;
    int * Pole=(int*)malloc(j*sizeof(int));
    //ukazatel na integer muzeme take : int * Pole; Pole=(int*)malloc(j*sizeof(int));
    printf("Zadej radu cisel, konec zadavani je CTRL+Z: ");
    while (!feof(stdin))
    {
        if ((scanf("%d", &(Pole[i++]))) != 1) && !feof(stdin)) //lze take: Pole + i
        {
            printf("Nespravny vstup\n");
            return 1;
        }
        if (i==j)
        {
            j*=2;
            Pole = realloc(Pole, j*sizeof(int));
        }
    }
    for (j = 0; j < i-1; j++) printf("%d, ", Pole[j]);
    return 0;
}
```

```
C:\Users\Správce\Desktop>a.exe
Zadej radu cisel, konec zadavani je CTRL+Z: 3
2
41
56
787
-5
-987
4
6
4
87
954
45
62
458
123
457
^Z
3, 2, 41, 56, 787, -5, -987, 4, 6, 4, 87, 954, 45, 62, 458, 123, 457,
```

Ukázka programu se vstupem do dynamického pole charů s realokací

```
#include <stdio.h>
#include <stdlib.h>

int main (void)
{
    int i=0, j=10;
    char * Pole=(char*)malloc(j*sizeof(char));
    printf("Zadej radu znaku: ");
    while (Pole[i-1]!='\n')           //konce zadavani je Enter
    {
        scanf("%c", &(Pole[i++]));
        if (i==j)
        {
            j*=2;
            Pole = realloc(Pole, j*sizeof(char));
        }
    }
    for (j = 0; j < i-1; j++) printf("%c, ", Pole[j]);
    return 0;
}
```




```
C:\Users\Správce\Desktop>a.exe
Zadej radu znaku: Pole charu.
P, o, l, e, , c, h, a, r, u, .,
```


Ukázka programu se vstupem z datového souboru

```
#include <stdio.h>

int main (void)
{
    int a, b;
    printf("Zadej dve cisla oddelena mezerou: ");
    if (scanf("%d %d",&a,&b)!=2)
    {
        printf("Chyba na vstupu!\n");
        return 1;
    }
    printf("Soucet cisel je %d.\n",a+b);
    return 0;
}
```

Datový soubor vytvoříme v poznámkovém bloku, zapíšeme do něj dvě čísla oddělená mezerou a uložíme se jménem souboru vstup.txt. Pozor, stejný formát vstupu musí být definovaný v příkazu scanf: scanf("%d %d",&a,&b) – integer, mezera, integer

 vstup.txt – Poznámkový blok

Soubor Úpravy Formát Zobrazení Nápověda

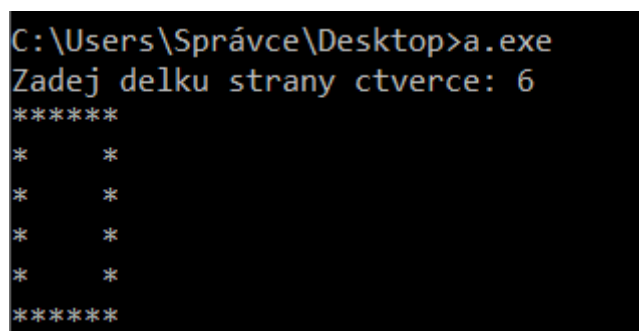
2 7

```
C:\Users\Správce\Desktop>a.exe <vstup.txt
Zadej dve cisla oddelena mezerou: Soucet cisel je 9.
```

Ukázka programu, který nakreslí čtverec

```
#include <stdio.h>

int main (void)
{
    int a,x,y;
    printf("Zadej delku strany ctverce: ");
    if (scanf("%d",&a)!=1)
    {
        printf("Nespravny vstup!");
        return 1;
    }
    for (y=1;y<=a;y++)
    {
        for (x=1;x<=a;x++)
        {
            if ((y==1)||(y==a))
            {
                printf("*");
            }
            else
            {
                if ((x==1)||(x==a)) printf("*"); else printf(" ");
            }
        }
        printf("\n");
    }
    return 0;
}
```



```
C:\Users\Správce\Desktop>a.exe
Zadej delku strany ctverce: 6
*****
*      *
*      *
*      *
*      *
*****
```

Čtverec zde znamená objekt se dvěma dvojicemi paralelních stran o stejném počtu hvězdiček!

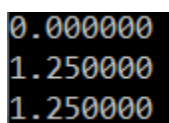
Přetypování

Někdy potřebujeme přetypovat hodnotu, například z integer na double. Typický příklad je, když dělíme hodnotami dvou integerů. Protože jsou obě čísla integer, výsledek je také integer! To je vlastnost jazyka C.

Například: jestli budeme dělit $5/4$, bude výsledek 1. Když výsledek přetypujeme na float, bude výsledek 1,25. Máme dvě možnosti:

- a) výsledku dáme typ float
- b) výsledek vynásobíme číslem 1.0

```
int i=5,j=4;
printf("%f\n",i/j);           //integer
printf("%f\n",1.0*i/j);      //float
printf("%f\n",(float)i/j);    //float
return 0;
```



```
0.000000
1.250000
1.250000
```

Ošetření vstupu

Vstupní hodnotu je nutné prověřit. Jestli došlo k chybě, máme dvě možnosti:

- a) ukončit program
- b) opakovat vstup

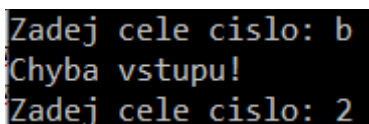
Pro potřeby našich domácích úkolů vždy budeme program ukončovat!

- a) ukončení programu

```
int i;
printf("Zadej cele cislo: ");
if (scanf("%d",&i)!=1)
{
    printf("Chyba!\n");
    return 1;
}
```

- b) opakování vstupu

```
int i;
printf("Zadej cele cislo: ");
while (scanf("%d",&i)!=1)
{
    printf("Chyba vstupu!\n");
    printf("Zadej cele cislo: ");
    fflush(stdin);           //vyčistí paměť pro vstup, když to neuděláme, chyba zůstane
}
```



```
Zadej cele cislo: b
Chyba vstupu!
Zadej cele cislo: 2
```

Práce s polem 2D (dvourozměrné pole)

Program, který udělá součet hodnot (čísel) v celém poli:
Pole je zadáno dynamicky.

```
#include <stdio.h>
#include <stdlib.h>

int VstupRadky (int * Radky)
{
    printf("Zadej pocet radku: ");
    return (scanf("%d",Radky) == 1) && (*Radky > 0);
}

int VstupSloupce (int * Sloupce)
{
    printf("Zadej pocet sloupce: ");
    return (scanf("%d",Sloupce) == 1) && (*Sloupce > 0);
}

int VstupCisla (int R, int S, int **Pole)
{
    int i,j;
    for (i=0;i<R;i++)
    {
        for (j=0;j<S;j++)
        {
            printf("Zadej cislo bunky [%d][%d]: ",j,i);
            if (scanf("%d",& Pole[j][i])!=1) return 1;
        }
    }
    return 0;
}

void Pocitej (int R, int S, int ** Pole)
{
    int i,j,suma=0;
    for(i=0;i<R;i++)
    {
        for(j=0;j<S;j++)
        {
            suma+=Pole[j][i];
        }
    }
    printf("Soucet cisel pole je %d\n",suma);
}

int main (void)
{
    int Sloupce, Radky, Kc;
    if (VstupSloupce(&Sloupce)==0){printf("Chyba!\n"); return 1;}
```

```

    if (VstupRadky(&Radky)==0){printf("Chyba!\n"); return 1;}
    int **Pole, **Vysledky,i; //deklarace ukazatele na ukazatel na integer
    Pole=(int**)malloc(Sloupce*sizeof(int*));
    for(i=0;i<Sloupce;i++) Pole[i]=(int*)malloc(Radky*sizeof(int));
    //muzeme take deklarovat staticke pole: int Pole[Sloupce][Radky]; potom musíme odstranit
    z parametru funkci **
    if (VstupCisla(Radky,Sloupce,Pole)==1){printf("Chyba!\n"); return 1;}
    Pocitej(Radky,Sloupce, Pole);
    free(Pole);
    return 0;
}

```

```

Zadej pocet sloupce: 4
Zadej pocet radku: 3
Zadej cislo bunky [0][0]: 2
Zadej cislo bunky [1][0]: 4
Zadej cislo bunky [2][0]: 5
Zadej cislo bunky [3][0]: 6
Zadej cislo bunky [0][1]: 1
Zadej cislo bunky [1][1]: 7
Zadej cislo bunky [2][1]: 8
Zadej cislo bunky [3][1]: 1
Zadej cislo bunky [0][2]: 0
Zadej cislo bunky [1][2]: 3
Zadej cislo bunky [2][2]: 7
Zadej cislo bunky [3][2]: 9
Soucet cisel pole je 53

```

Práce s polem 2D (dvourozměrné pole)

Program, který udělá součet hodnot (čísel) v oblasti poli, která je definována koordináty levého horního a pravého spodního rohu oblasti (od 0 do Sloupce-1 a Radky-1) :

Pole je zadáno dynamicky.

Vše je řešeno funkcemi.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int VstupRadky (int * Radky)
{
    printf("Zadej pocet radku: ");
    return (scanf("%d",Radky) == 1) && (*Radky > 0);
}
```

```
int VstupSloupce (int * Sloupce)
{
    printf("Zadej pocet sloupce: ");
    return (scanf("%d",Sloupce) == 1) && (*Sloupce > 0);
}
```

```
int VstupPozice (int * X, int * Y, int S, int R, int Misto)
{
    printf("Zadej souradnice %d. pozice oddelene carkou 0..S-1,0..R-1: ", Misto);
    return ((scanf("%d, %d",X,Y) == 2) && (*X >= 0) && (*X < S) && (*Y >= 0) && (*Y <
R));
}
```

```
int VstupCisla (int R, int S, int **Pole)
{
    int i,j;
    for (i=0;i<R;i++)
    {
        for (j=0;j<S;j++)
        {
            printf("Zadej cislo bunky [%d][%d]: ",j,i);
            if (scanf("%d",& Pole[j][i])!=1) return 1;
        }
    }
    return 0;
}
```

```

void Pocitej (int R, int S, int ** Pole, int X1, int Y1, int X2, int Y2)
{
    int i,j,suma=0;
    for(i=Y1;i<=Y2;i++)
    {
        for(j=X1;j<=X2;j++)
        {
            suma+=Pole[j][i];
        }
    }
    printf("Soucet cisel pole je %d\n",suma);
}

```

```

void Tisk (int R, int S, int ** Pole)
{
    int i,j;
    for(i=0;i<R;i++)
    {
        for(j=0;j<S;j++)
        {
            printf("%d, ",Pole[j][i]);
        }
        printf("\n");
    }
}

```

```

int main (void)
{
    int Sloupce, Radky, X1, X2, Y1, Y2;
    if (VstupSloupce(&Sloupce)==0){printf("Chyba!\n"); return 1;}
    if (VstupRadky(&Radky)==0){printf("Chyba!\n"); return 1;}
    if (VstupPozice(&X1,&Y1,Sloupce,Radky,1)==0){printf("Chyba!\n"); return 1;}
    if (VstupPozice(&X2,&Y2,Sloupce,Radky,2)==0){printf("Chyba!\n"); return 1;}
    if ((X1>X2)|| (Y1>Y2)){printf("Chyba 3!\n"); return 1;}
    int **Pole, **Vysledky,i; //deklarace ukazatele na ukazatel na integer
    Pole=(int**)malloc(Sloupce*sizeof(int*));
    for(i=0;i<Sloupce;i++) Pole[i]=(int*)malloc(Radky*sizeof(int));
    //muzeme take deklarovat staticke pole: int Pole[Sloupce][Radky]; potom musíme odstranit
    z parametru funkci **
    if (VstupCisla(Radky,Sloupce,Pole)==1){printf("Chyba!\n"); return 1;}
    Tisk(Radky,Sloupce,Pole);
    Pocitej(Radky,Sloupce, Pole, X1, Y1, X2, Y2);
    free(Pole);
    return 0;
}

```

```
Zadej pocet sloupcu: 4
Zadej pocet radku: 3
Zadej souradnice 1. pozice odddelene carkou 0..S-1,0..R-1: 1,1
Zadej souradnice 2. pozice odddelene carkou 0..S-1,0..R-1: 2,2
Zadej cislo bunky [0][0]: 2
Zadej cislo bunky [1][0]: 1
Zadej cislo bunky [2][0]: 3
Zadej cislo bunky [3][0]: 0
Zadej cislo bunky [0][1]: 4
Zadej cislo bunky [1][1]: 5
Zadej cislo bunky [2][1]: 1
Zadej cislo bunky [3][1]: 2
Zadej cislo bunky [0][2]: 3
Zadej cislo bunky [1][2]: 7
Zadej cislo bunky [2][2]: 4
Zadej cislo bunky [3][2]: 5
2, 1, 3, 0,
4, 5, 1, 2,
3, 7, 4, 5,
Soucet cisel pole je 17
```