

UPR

Cvičení 2

Funkce

- Pojmenovaný blok kódu, který lze parametrizovat
 - může vracet hodnotu (výjimkou je typ `void`, který nic nevrací)

Syntaxe:

```
<návratový typ> <název> (<typ1> <prom1>, ... ) {  
    <kód funkce>  
}
```

- Již známe :

`int main()` - vrací celé číslo (proto `return 0` na konci), nemá žádné parametry

Funkce

```
int secti(int a, int b) {  
    return a + b;  
}
```

funkci pak můžeme zavolat

```
int x = 10;  
int suma = secti(x, 20 + x);  
printf("Vysledek je %d\n", suma);
```

Funkce

- Proměnné vytvořené ve funkci existují pouze v rámci funkce!

```
int funkce(int a)
```

```
{
```

```
    int b = a*2;
```

```
} // zde funkce konci a promenna b zanika.
```

Pokud ji chceme pouzivat i jinde, musime ji vratit pomoci return

Funkce

- A také, parametry jsou předané hodnotou, tzn. že změna se projeví pouze uvnitř funkce a ne mimo ni! Jak to vyřešit, si ukážeme později.

```
void funkce(int a) {
    a = a*2;
    printf("Hodnota a ve funkci: %d\n", a); // 10
}

int main() {
    int a = 5;
    funkce(a);
    printf("Hodnota a mimo funkci: %d\n", a); // 5
    return 0;
}
```

Úkol

- Minule jste měli úkol vypsát pomocí cyklu sudá čísla od 10 do 0 a spočítat jejich sumu
- Upravte úkol tak, že tento kód přesunete do funkce a tato funkce vrátí onu sumu čísel
- Parametrem funkce bude hodnota, od které se bude počítat (tzn. ne od čísla 10, ale od nějakého zadaného čísla > 0)
- V mainu zavolejte funkci a vytiskněte sumu

Podmínky

- Máme více podmínek zároveň
chceme, at' platí obě podmínky zároveň:

```
if ( podminka1 && podminka2 )  
{...
```
- chceme, at' platí alespoň jedna podmínka:

```
if ( podminka1 || podminka2 )  
{...
```
- chceme, at' platí kombinace více podmínek (&& má přednost před || podobně jako násobení před sčítáním):

```
if ( podminka1 || podminka2 && podminka3 )  
{...
```

Podmínky

- Datový typ bool
Je třeba vložit hlavičku `stdbool.h`

```
#include <stdbool.h>
bool podminka1 = true;
bool podminka2 = false;
bool podminka3 = 3 > 2;
bool podminka4 = !podminka1;
```


Podmínky - úkol

- Najděte maximum ze tří desetinných čísel zadaných uživatelem

```
float a, b, c;  
printf("zadejte 3 cisla: ");  
scanf("%f", &a);  
scanf("%f", &b);  
scanf("%f", &c);  
// zde pokračujte svym kodem, muzete pouzit  
funkci, ktera najde maximum ze tri zadaných  
parametru (bude float misto int)
```

Cyklus for

- Minule jsme si ukázali cyklus while

```
while (<podminka>) { ...
```

```
int a = 0;
while(a <= 10)
{
    printf("%d\n", a);
    a++;
}
```

Cyklus for

- Známe počet opakování?

- Použijme cyklus for

```
for (<inicializace>; <podminka>; <inkrement>)
```

```
for (int a = 0; a <= 10; a++)  
{  
    printf ("%d\n", a);  
}
```

Cyklus for - úkol

- Vypište pomocí cyklu `for` druhé mocniny čísel od 0 po číslo `x`, které zadá uživatel