

Parametry funkcí

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.



Palacký University, Olomouc



- počet argumentů a jejich typy předány formátovacím řetězcem (např. `printf`)
- předpokládáme typ parametrů a funkci je předán jejich počet (např. `suma`, `průměr`)
- předpokládáme typ parametrů a mám určenou zarážku (např. práce s řetězci)



- ... výpustka
- `typ funkce(typ_1 p1, typ_2 p2, ...);`

Example

```
double prumer(int pocet, ...);  
prum = prumer(5, 1.0, 2.5, 3.1, 0.8, 0.0);
```



- knihovna `stdarg.h`
- typ: `va_list`
- makra:
 - `va_start()`
 - `va_arg()`
 - `va_end()`

Example

```
va_list parametry;  
  
va_start(parametry, posledni_povinky);  
  
promenna = va_arg(parametry, typ);  
  
va_end(parametry);
```

- 1 Napište funkci, která vrátí nejmenší z předaných celočíselných parametrů. Funkce bere jako první argument počet předaných celých čísel.
- 2 Napište funkci `my_printf()`, která se bude chovat obdobně jako funkce `printf()`. Bude mít jeden povinný argument – řetězec, který může obsahovat formátovací sekvence, a libovolný počet nepovinných argumentů.

Formátovací sekvence:

- `*i` nahradí celým číslem, který byl předán jako nepovinný argument,
- `*c` nahradí znakem,
- `*f` nahradí desetinným číslem.

Pozor! funkce `va_arg()`, pracuje jen se základními datovými typy. Znak (`char`), je potřeba načíst jako `int` a přetypovat na `char`

```
(char)va_arg(parametry, int)
```

a desetinná čísla je potřeba načíst jako typ `double`

```
va_arg(parametry, double)
```

Uvnitř funkce je možné použít funkci `printf()`.



```
int main(int argc, char *argv)
```

- **argc** = argument count
- **argv** = argument vector

- `argv[0]` název programu

Example

```
int main(int argc, char* argv[])
{
    int i;

    for(i=0;i<argc; i++)
    {
        printf("%s\n", argv[i]);
    }
    return 0;
}
```

```
Príkazový řádek
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\marke>cd ../../Skola\vyuka2020-2021\LS\ZP2\kody

C:\Skola\vyuka2020-2021\LS\ZP2\kody>pr7-priklad.exe parametr1 parametr2
pr7-priklad.exe
parametr1
parametr2

C:\Skola\vyuka2020-2021\LS\ZP2\kody>_
```

- 1** Běžnou konvencí jsou volitelné argumenty začínající `-`.
Upravte předchozí výpis tak, aby se vypsaly jednotlivé argumenty (kromě názvu programu) každý na jeden řádek. Pokud jako první bude uveden volitelný argument `-n` vypíšíou se i čísla řádků (tento argument se nevypíše).
- 2** Modifikujte předchozí příklad tak, aby rozlišoval volitelné argumenty `-n` (vypíše čísla řádků) a `-o` (přidá před každý argument odřádku).
- 3** Napište program `soucet`, který sečte čísla zadaná v příkazové řádce. Každé číslo je samostatný argument. Například: `soucet 2 3 4`
vrátí 9
- 4** Napište program `vyraz` vyhodnocující výpočty zapsané v obrácené polské notaci a zadané z příkazové řádky, kde každý operand nebo operátor je samostatným argumentem. Například: `vyraz 2 3 4 + *`
vypočítá výraz $2 \cdot (3 + 4)$
- 5** Napište program `nejdelssi`, který vrátí nejdelší (obsahující nejvíce znaků) ze svých argumentů. Například: `nejdelssi ahoj jak se mas`
vypíše `ahoj`