



KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLMOUCI

Parametry funkcí

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.

Funkce s proměnným počtem argumentů

- 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)

Deklarace

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

Příklad

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

- parametry na zásobníku za sebou

Zpracování parametrů

- `stdarg.h`
- typ `va_list`, který se používá k uložení parametrů v zásobníku,
- makro `va_start()`,
- makro `va_arg()`,
- makro `va_end()`.

```
va_list parametry;  
va_start(parametry, posledni_povinny);
```

```
cislo = va_arg(parametry, double);
```

```
va_end(parametry);
```

Funkce s proměnlivým počtem argumentů

Příklad

Příklad

Napište funkci průměr, která bere jako první argument celé číslo představující počet předávaných nepovinných argumentů. Ve funkci spočítejte průměr těchto čísel.

Příklad

Příklad (pr7-prumer.c)

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

double prumer(int pocet, ...){
    double soucet = 0;
    int i;
    va_list parametry;
    va_start(parametry, pocet);
    for(i = 0; i < pocet; i++){
        soucet += va_arg(parametry, double);
    }
    va_end(parametry);
    return soucet / pocet;
}

int main(){
    printf("%f \n", prumer(5, 1.2, 3.4, 5.6, 7.8, 9.0));
    return 0;
}
```

Cvičení

- 1 Napište funkci, která vrací 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()`.

Parametry příkazové řádky

- `argc` (argument count) – počet argumentů s kolika je příkaz spuštěn,
- `argv` (argument vector) – ukazatel na pole znakových řetězců obsahujících argumenty. Vždy jeden parametr v jednom řetězci.
- `argv[0]` – název programu

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


Parametry příkazové řádky

Příklad

- Součet dvou čísel – `soucet.exe`
- Spuštění v konzoli – `soucet.exe 2 3`
- `argc = 3`
- `argv`:
 - `argv[0] = "soucet.exe"`
 - `argv[1] = "2"`
 - `argv[2] = "3"`

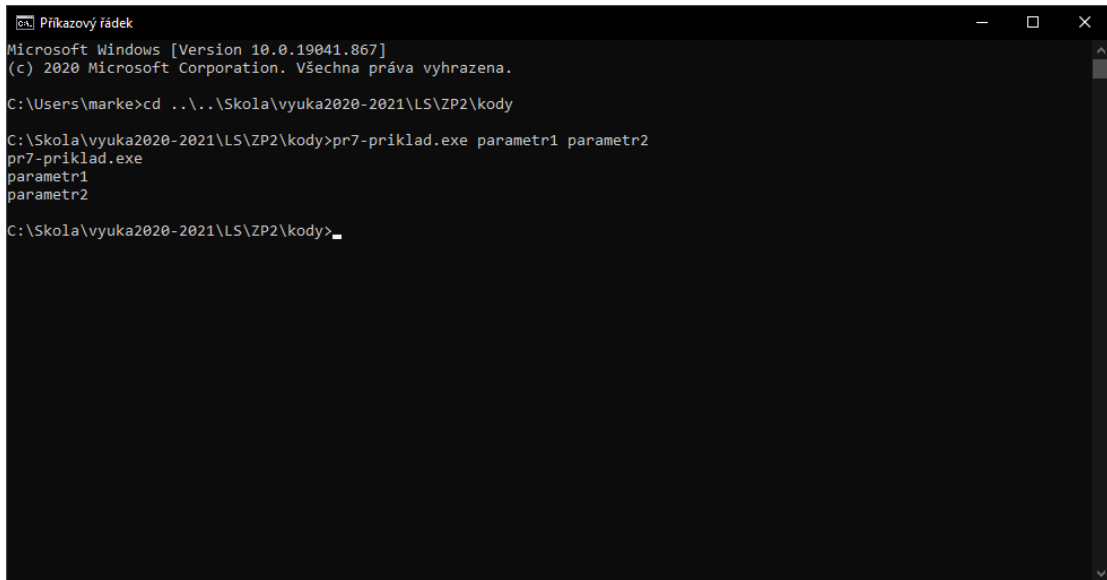
Konzole

Příklad

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

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

Konzole



```
Příkazový 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>_
```

Cvičení

- 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` vypíšou 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 odrážku).
- 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