

Paměť 2

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.



Palacký University, Olomouc

- **předávání argumentů funkci:**

- hodnotou
- odkazem = předáváme jeho adresu

Example

```
// navratovy typ funkce: adresa typu int
int deleni(int a, int b, int *r)
{
    /* r – zbytek po deleni */
    *r = a%b;
    return a/b;
}

int main()
{
    int x,y;
    /* x = 2, y = 3*/
    x = deleni(13,5,&y);
    return 0;
}
```

- `#include <string.h>`
- `void *memcpy(void *mem1, void *mem2, size_t n);`

Example

```
/* spatne. kopirujeme pouze adresu */  
foo = bar;  
  
/* kopie cyklem */  
for(int i = 0; i < m; i = i+1)  
    foo[i] = bar[i];  
  
/* kopie pomoci memcpy */  
memcpy(foo, bar, sizeof(int)*m);
```

- `#include <string.h>`
- `void *memmove(void *mem1, void *mem2, size_t n);`

Example

```
/* cyklem */  
for(int i = 0; i < m; i = i+1)  
    foo[i-1] = foo[i];  
  
/* pomoci memmove */  
memmove(foo, foo+1, sizeof(int)*(m-1));
```

- `#include <string.h>`
- `void *memset(void *mem, int c, size_t n);`

Example

```
/* cyklem */  
for(int i = 0; i < m; i = i+1)  
    foo[i] = 0;  
  
/* pomoci memset */  
memset(foo, 0, sizeof(int)*m);
```

- 1 Naprogramujte funkci pro spojení dvou textových řetězců. Argumentem funkce jsou dva textové řetězce ke spojení, funkce vrátí nový řetězec (jako ukazatel).
- 2 Naprogramujte funkci, která jako argument bere pole čísel a vrátí pole jejich druhých mocnin (tedy prvek na indexu i vráceného pole bude mít hodnotu druhé mocniny prvku na indexu i vstupního pole).
- 3 Naprogramujte funkci, která bere jako argument 2 proměnné a provede výměnu hodnot těchto dvou proměnných.
- 4 Napište funkci, která pro 2 zadaná čísla vrátí, zda je možné udělat jejich podíl a pokud ano, vrátí i jejich podíl.
- 5 Napište funkci, která bere 2 argumenty (text a podretezec). Funkce v daném textovém řetězci text vyhledá první výskyt zadaného podřetězce podretezec. Funkce vrátí ukazatel na první znak nalezeného podřetězce nebo konstantu NULL, pokud podřetězec hledany nebyl nalezen.
- 6 Napište funkci `int porovnej`, která porovná 2 předané textové řetězce a vrátí -1, pokud je první řetězec menší než druhý, 0, pokud jsou řetězce shodné, nebo 1, pokud je druhý řetězec menší než první. Při práci s textovými řetězci používejte výhradně ukazatele, operátor dereference a pointerovou aritmetiku.

- 7** (Úloha za 5 bodů, pokud řešení zašlete na marketa.trneckova@gmail.com do dalšího cvičení) Napište funkci, která převede číslo na jeho římskou reprezentaci. Pravidla pro tvorbu číslic:
- Římská čísla se zapisují kombinací znaků I, V, X, L, C, D, M.
 - Římská čísla se skládají psaním od znaků pro nejvyšší hodnoty k nejnižším (MDL = 1550). Většinou se kombinují nejvýše 3 stejné římské číslice (XXX = 30, III = 3).
 - Menší římská číslice před větší znamená odečet (IV = 4). Takto se odečítá jen jediná římská číslice (jen zcela ojediněle dvě stejné číslice: IIX = 8).
 - Pro odčítání podle bodu 3 se používají pouze římské číslice I, X, C; v matematickém kontextu zcela výjimečně také M. Nikdy nebyly používány římské číslice V, L, D (správně: XC = LXXXX = 90, MCM = 1900; nesprávně: VC = 95).
 - Číslice I se pro odečítání většinou užívá jen před V, X. Není tedy správně MIM pro 1999 (lépe je MCMXCVIII nebo MCMXCIX).