

Zadání zápočtové úlohy z předmětu KMI/XJC

Úlohu odevzdejte nejpozději do 8. 12. 2023 (do 23:59) přes MS Teams, případně zašlete na e-mail **marketa.trneckova@gmail.com** s předmětem „KMI/XJC započetova uloha“.

Naprogramujte modul (bez funkce `main()`) pro práci s polynomy (potřebná teorie je na konci zadání). Modul se bude skládat ze zdrojového kódu (soubor `polynomy.c`) a z hlavičkového souboru (soubor `polynomy.h`).

V modulu musí být definován strukturovaný datový typ `polynom`. Struktura si bude uchovávat informaci o stupni polynomu (`int stupen`) a koeficientech u jednotlivých mocnin proměnné polynomu – od 0. Koeficienty budou uloženy v poli, které bude mít počet prvků roven `stupen + 1` (`int* koeficienty`).

Dále modul obsahuje následující funkce:

- `polynom vytvor_z_retezce(char* retezec)` – funkce vytvoří polynom, který je zadán řetězcem ve tvaru „`a0 a1 a2 ... an`“. `a0, ..., an` jsou posloupnosti číslic (záporné koeficienty začínají znakem `-`). `a0` představuje koeficient u nulté mocniny, `a1` koeficient u 1. mocniny, atd.
- `polynom* vytvor_ze_souboru(char *soubor, int *pocet)` – funkce vytvoří pole polynomů, které jsou uloženy v souboru `soubor` (řetězec obsahující cestu k souboru). Každý polynom je na samostatném řádku a je zadán řetězcem ve tvaru „`a0 a1 a2 ... an`“. `a0, ..., an` jsou posloupnosti číslic (záporné koeficienty začínají znakem `-`). `a0` představuje koeficient u nulté mocniny, `a1` koeficient u 1. mocniny, atd. Do proměnné, která je předána odkazem `pocet` uloží počet načtených polynomů.
- `polynom vytvor(int stupen, ...)` – variadická funkce vytvoří polynom, který má stupeň `stupen` a koeficienty (typ `int`) jsou dány nepovinnými argumenty, kterých je `stupen + 1`.
- `polynom soucet(polynom p1, polynom p2)` – funkce vrátí součet polynomů `p1` a `p2`.
- `polynom rozdil(polynom p1, polynom p2)` – funkce vrátí rozdíl polynomů `p1` a `p2`.
- `polynom soucin(polynom p1, polynom p2)` – funkce vrátí součin polynomů `p1` a `p2`.
- `float vycisli(polynom p, float x)` – funkce vrátí reálné číslo, které získáme dosazením `x` do polynomu `p`. Pro polynom $3 + x + x^2$ a číslo 1.0 vrátí 5.0.
- `void vypis(polynom p)` – funkce vypíše polynom do konzole ve tvaru „`a0 + a1x + a2x2 + a3x3 + ... + anxn`“. `ai` jsou koeficienty (celá čísla). V případě, že je koeficient roven 0, celý člen se nevypíše, pokud je roven 1 vypíše se pouze `xn`. Pokud je koeficient záporný, nevypíše plus, ale mínus (`-`). Viz následující výpisy.

$5 + 2x + 3x^2$ pro polynom $5 + 2x + 3x^2$
 $5 + x$ pro polynom $5 + x$
 $5 - 3x + 5x^2$ pro polynom $5 - 3x + 5x^2$
 $5 + x^2$ pro polynom $5 + x^2$

- `void vypis_do_souboru(polynom *pole_polynomu, char* soubor, int pocet)` – uloží pole polynomů `pole_polynomu` do souboru `soubor` (řetězec s cestou k souboru). Každý polynom uloží na samostatný řádek ve tvaru, v jakém se polynomy vypisují do konzole. V proměnné `pocet` je uložena informace o počtu polynomů v poli.
- `void smaz(polynom p)` – funkce uvolní paměť alokovanou pro pole koeficientů polynomu `p`.
- `int je_koren(polynom p, float x)` – funkce vrátí 1 pokud je `x` kořenem polynomu `p`, 0 pokud není.

Dále obsahuje:

- Globální proměnnou `chyba`, která bude uchovávat, zda v poslední operaci nedošlo k nějaké chybě.
- Definujte symbolické konstanty pro různé chyby:
 - `BEZ_CHYBY` – pokud ve funkci nedošlo k žádné chybě.
 - `CHYBA_ALOKACE` – chyba při alokaci paměti.
 - `CHYBA_OTEVRENI` – chyba při otevírání souboru.
 - `CHYBA_ZAVRENI` – chyba při zavírání souboru.
 - `CHYBA_TVARU` – chyba pokud polynom není ve správném tvaru.
 - `CHYBA_JINA` – ostatní chyby.

Modul může obsahovat i další pomocné funkce.

Opisování NETOLERUJI! Raději konzultujte se mnou, než se spolužáky.

V případě jakýchkoliv nejasností se nebojte se na mě obrátit.

Je možné použít knihovny: `stdlib.h`, `stdio.h`, `stdarg.h`

Příklad použití

Pro následující zdrojový kód

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

#include "polynomy.h"

int main(){
    polynom p1, p2, p3, p4;
    polynom *pole_polynomu;
    int pocet, i;

    p1 = vytvor(2,5,2,3);

    printf("Polynom p1: \n");
    vypis(p1);
    printf("\n\n");

    printf("Polynom p1(2): %f", vycisli(p1,2.0));
    printf("\n\n");

    p2 = vytvor_z_retezce("1 -1 -5");
    printf("Polynom p2: \n");
    vypis(p2);
    printf("\n\n");

    p3 = soucet(p1, p2);
    printf("Soucet polynomu p1 + p2: \n");
    vypis(p3);
    printf("\n\n");

    p4 = rozdil(p1, p2);
    printf("Rozdil polynomu p1 - p2: \n");
    vypis(p4);
    printf("\n\n");

    pole_polynomu = vytvor_ze_souboru("neexistujici.txt", &pocet);
    printf("Polynomy: \n");
    switch(chyba){
        case CHYBA_OTEVRENI:
            printf("Nepodarilo se otevrit soubor.");
            break;
        case CHYBA_ALOKACE:
            printf("Nastala chyba alokace pameti.");
            break;
        case BEZ_CHYBY:
            for(i = 0; i<pocet; i++)
                vypis(pole_polynomu[i]);
            break;
    }
    printf("\n\n");

    return 0;
}
```

dostaneme tento výstup

```
Polynom p1:  
5 + 2x + 3x2  
  
Polynom p1(2): 21.000000  
  
Polynom p2:  
1 - x - 5x2  
  
Soucet polynomu p1 + p2:  
6 + x - 2x2  
  
Rozdil polynomu p1 - p2:  
4 + 3x + 8x2  
  
Polynomy:  
Nepodarilo se otevrit soubor.  
  
Process returned 0 (0x0)   execution time : 0.035 s  
Press any key to continue.
```

V kódu kvůli jeho délce chybí detekce chyb při vykonávání většiny operací.

Teorie

Polynom – Polynom je výraz ve tvaru $p(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$. a_i nazýváme *koeficienty polynomu*. $a_n \neq 0$. Pro jednoduchost budeme v našem případě předpokládat, že jsou koeficienty celočíselné.

Stupeň polynomu – nejvyšší exponent proměnné x s nenulovým koeficientem. Stupeň polynomu $5 + 2x + 3x^2$ je roven 2.

Součet polynomů – $f(x) = \sum_{i=0}^n a_i x^i$, $g(x) = \sum_{i=0}^m b_i x^i$. Součet

$$h(x) = \sum_{i=0}^{\max(m,n)} (a_i + b_i) x^i$$

Rozdíl polynomů – $f(x) = \sum_{i=0}^n a_i x^i$, $g(x) = \sum_{i=0}^m b_i x^i$. Rozdíl

$$h(x) = \sum_{i=0}^{\max(m,n)} (a_i - b_i) x^i$$

Součin polynomů – získáme vzájemným vynásobením jednotlivých koeficientů.

$$h(x) = \sum_{i=0}^n a_i x^i \cdot \sum_{i=0}^m b_i x^i = \sum_{i=0}^{m+n} \left(\sum_{j=0}^i (a_j \cdot b_{i-j}) \right) x^i.$$

Kořen polynomu – číslo α se nazývá kořen, pokud platí $p(\alpha) = 0$.