



UNIVERSIDADE DE SÃO PAULO - ICMC

Departamento de Ciências de Computação e Estatística

SCC 202 - Algoritmos e Estruturas de Dados I- TURMA B - 2º Sem /2010

PROFA: Sandra Aluisio

WIKI: <http://wiki.icmc.usp.br/index.php/SCC-202>

Projeto 2: Manipulação de Polinômios Esparsos

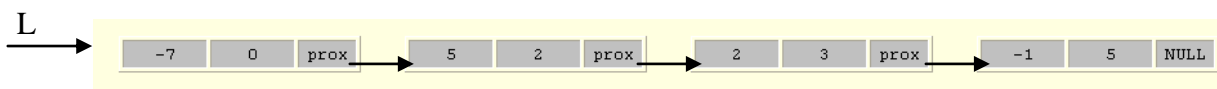
Data de entrega: 9/11

Este trabalho trata da manipulação de polinômios (numa mesma variável) com coeficientes inteiros. Um polinômio é constituído por um conjunto de termos. Cada um destes termos é constituído por um número inteiro positivo (incluindo o zero) N que representa o grau e um número inteiro K que representa o coeficiente. Supondo que temos um termo na variável x , a representação matemática desse termo é:

$$Kx^N$$

Cada polinômio pode ser armazenado em uma LISTA. Cada nó da lista armazena os dados de um dos termos do polinômio: o coeficiente e o expoente, além do apontador para o próximo termo. Apenas termos com coeficiente não-nulo devem estar na lista.

Exemplo: O polinômio $p(x) = -x^5 + 2x^3 + 5x^2 - 7$ **poderia** ser armazenado em uma lista L de acordo com a seguinte figura:



Há outras possibilidades de representação que devem ser escolhidas pelo grupo (vejam comentário no fim do trabalho).

Implementem um **TAD polinômio** para representar polinômios com as seguintes operações (polinômio.h):

```
typedef struct termo *Pol;
```

```
Pol newPol(int);
```

Cria um novo polinômio. Recebe um inteiro que indica o maior grau possível para este polinômio.

```
void destroyPol(Pol);
```

Apaga um polinômio, liberando sua memória.

```
void addTerm(Pol, int, int);
```

Junta um termo a um polinômio. Recebe dois inteiros, respectivamente, o grau do termo e o coeficiente.

```
void delTerm(Pol, int);
```

Apaga um termo de um polinômio. Recebe um inteiro que indica o grau do termo a ser apagado.

```
int maxExp(Pol);
```

Retorna o grau do polinômio. O grau de um polinômio é o do termo de maior grau.

```
int coeff(Pol, int);
```

Retorna o coeficiente de um termo de um polinômio. O inteiro representa o grau do termo pretendido.

Pol sumPol (Pol, Pol) ;

Somar dois polinômios corresponde a somar os termos de igual grau de cada um dos polinômios: $k_1x^n + k_2x^n = (k_1 + k_2)x^n$. Retorna um novo polinômio.

Pol subPol (Pol, Pol) ;

Subtrair dois polinômios corresponde a subtrair os termos de igual grau de cada um dos polinômios: $k_1x^n - k_2x^n = (k_1 - k_2)x^n$. Retorna um novo polinômio.

Pol mulPol (Pol, Pol) ;

Para multiplicar dois polinômios usa-se a propriedade distributiva: multiplica-se um polinômio pelos termos do outro. Retorna um novo polinômio.

Pol derPol (Pol) ;

Deriva um polinômio. Esta operação corresponde à derivação dos seus termos, melhor explicada abaixo. Retorna um novo polinômio.

float evalPol (Pol, float) ;

Calcula o valor do polinômio quando a variável assume um dado valor (que é passado para a função como real).

void printPol (Pol) ;

Escreve o polinômio na tela.

OBS: Os grupos devem incluir os parâmetros para indicar possíveis erros nas operações!

Notem que os polinômios não podem ter mais do que um termo do mesmo grau, porque eles podem ser reduzidos a um só termo. Por exemplo, se os dois termos:

$k_1 x^n + k_2 x^n$ pertencem a um mesmo polinômio, então podem ser representados pelo termo: $(k_1 + k_2) x^n$

(I) A derivada de um polinômio corresponde à soma das derivadas dos seus termos. Portanto, se $p(x) = f_m(x) + f_{m-1}(x) + \dots + f_1(x) + f_0(x)$ e:

$f_m(x) = a_m x^m$	$f_{m-1}(x) = a_{m-1} x^{m-1}$... = ...	$f_1(x) = a_1 x^1$	$f_0(x) = a_0 x^0$
--------------------	--------------------------------	-----------	--------------------	--------------------

Então $p'(x) = f'_m(x) + f'_{m-1}(x) + \dots + f'_1(x) + f'_0(x)$, onde $p'(x)$ corresponde à derivada do polinômio $p(x)$. Perceba que $p'(x)$ é também um polinômio.

(II) Se $f(x) = b x^m$, então $f'(x)$ será definida conforme abaixo:

(i) Se $m \neq 0$, $f'(x) = m \cdot b x^{m-1}$

(ii) Se $m = 0$, $f'(x) = 0$.

Onde $f'(x)$ é a derivada de $f(x)$.

Para resolver o trabalho, o grupo deve implementar o TAD Polinômio com base nas operações de LISTAS escolhendo entre:

- REPRESENTAÇÃO: SEQUENCIAL ou ENCADEADA
- IMPLEMENTAÇÃO: ESTÁTICA ou DINÂMICA

Escolha uma das marcadas em negrito abaixo:

Rep/Imp	ESTÁTICA	DINÂMICA
SEQUENCIAL	SEQUENCIAL ESTÁTICA	SEQUENCIAL DINÂMICA
ENCADEADA	ENCADEADA ESTÁTICA	ENCADEADA DINÂMICA

Além disso, há ainda as opções: a) Lista Ordenada ou não Ordenada; b) Lista Circular ou não, c) Listas Duplamente Encadeadas, d) Listas com Nó Cabeça e as várias possíveis combinações entre (a), (b), (c), (d). **JUSTIFIQUE A ESCOLHA da ED.**

Programa Cliente

Devem implementar um programa cliente que permita efetuar todas as operações referentes aos polinômios descritas no TAD polinômio. A forma de interação do programa fica ao critério de cada grupo, sendo, no entanto, importante que o programa seja versátil. Sugere-se a utilização de um menu. Uma possível saída para algumas operações é:

```
(x^2-x^3+4x^5) + (-1+2x^2+x^3+3x^4) = -1+3x^2+3x^4+4x^5
O grau de -1+3x^2+3x^4+4x^5 e' 5.
(-1+3x^2+3x^4+4x^5) - (-1+2x^2+x^3+3x^4) = x^2-x^3+4x^5
(x^2-x^3+4x^5) * (x) = x^3-x^4+4x^6
A derivada de x^3-x^4+4x^6 e' 3x^2-4x^3+24x^5.
(3x^2-4x^3+24x^5) * (x-x^2) = 3x^3-7x^4+4x^5+24x^6-24x^7
```

Os trabalhos serão avaliados de acordo com os seguintes critérios:

- 1) **Corretude** do programa e do TAD: o programa deve fazer o que foi especificado;
- 2) Usabilidade da **interface**: a interface com o usuário deve ser clara, flexível e intuitiva;
- 3) Estruturas de dados utilizadas: **adequação e eficiência**;
- 4) Observação das **boas práticas** da programação de acordo com o GUIA DE CODIFICAÇÃO - Manual de Boas Práticas de Programação, preparado pela monitora PAE Arineiza Pinheiro.

Os trabalhos devem ser implementados em C, utilizando-se, preferencialmente, o ambiente Code-Blocks. Dizer qual o ambiente/compiler utilizado no caso da escola recair sobre outro. A cada dia de atraso, 1 ponto é descontado da nota, até 5 dias de atraso. Depois deste prazo o trabalho não será aceito.

Todos os seus arquivos .h ou .c devem ter um cabeçalho como o seguinte:

```
/* **** */
/* Aluno: Fulano de Tal */
/* Número USP: 12345678 */
/* Curso: ... */
/* Trabalho 2 -- Manipulação de Polinômios Esparsos */
/* SCC 202 -- 2010 -- ICMC/USP, turma XX -- Prof. YYYY */
/* Compilador: ... (gcc ou Code::Blocks) versão ... */
/* Sistema Operacional: ... */
/* **** */
```

A entrega dos projetos será em 2 PARTES:

1. Um arquivo zip/rar (Grupo_ALUNO1_ALUNO2.zip) para o e-mail da disciplina: scc202.turmab2010@gmail.com com: (a) arquivos de código-fonte do programa e TAD, (b) arquivo executável do programa (c) e o arquivo “.gcv” gerado para o Critério Todos-nós;

2. Documentação externa do programa de, no máximo, 5 páginas, contendo, pelo menos:
 - (a) breve descrição do trabalho, com a estrutura de dados utilizada com **JUSTIFICATIVA DA ESCOLHA da ED**,
 - (b) os casos de testes, seus comentários, telas de saída, bem como a porcentagem total de linhas executadas (gerado no console pela ferramenta Gcov),
 - (c) toda a documentação usual dos TADs (Entrada, Saída, pré e pós-condições, além dos comentários das rotinas) e
 - (d) uma seção detalhando como compilar e rodar o programa.

<p>A documentação deverá ser entregue na sala de aula ou na sala da professora até as 18:00 (se a profa não estiver, coloque embaixo da porta e envie e-mail; você receberá resposta confirmando entrega)</p>
