

Segundo Exercício-Programa

Calculadora de polinômios esparsos - Parte I

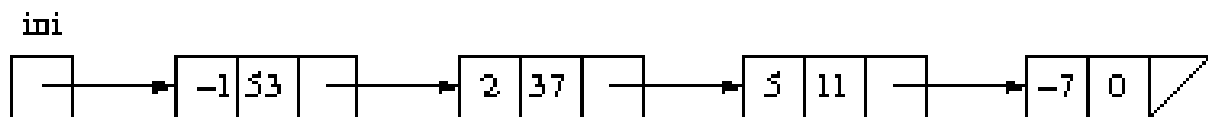
Entrega: 4 de outubro



O objetivo deste exercício-programa é exercitar o uso de listas encadeadas. O resultado deste EP e do próximo será uma calculadora que manipula polinômios esparsos.

Polinômios esparsos são polinômios usualmente de grau alto, mas com poucos termos não nulos. Tais polinômios podem ser armazenados eficientemente em uma lista encadeada que contém, em cada célula, os dados de um dos termos não nulos do polinômio.

Exemplo: O polinômio $p(x) = -x^{53} + 2x^{37} + 5x^{11} - 7$ poderia ser armazenado em uma lista de acordo com a seguinte figura:



onde ini é o apontador para o início da lista, o primeiro campo de cada célula guarda o coeficiente de um termo do polinômio, o segundo campo guarda o expoente do termo, e o terceiro campo é o usual apontador para a próxima célula da lista. Note a ausência de termos cujo coeficiente é nulo.

Na lista encadeada acima, não usamos cabeça de lista, mas você pode optar por usar, se isso tornar a sua implementação melhor. Observe que, na lista acima, as células aparecem em ordem decrescente de expoente. Esta também é uma opção do programador. As células poderiam aparecer em uma ordem arbitrária, ou em ordem crescente de expoente, ou numa outra ordem. A escolha é do programador, ou seja, é sua. Escolha a ordem que tornar a sua implementação a melhor possível. O único ponto que você não pode abrir mão na representação de polinômios esparsos é que a lista não deve conter termos com coeficiente nulo.

Para fazer a escolha do tipo de lista encadeada que você vai usar no seu programa, é essencial pensar nas operações que você vai implementar sobre estas listas. Vamos então a elas.

Operações da calculadora

As seguintes operações devem estar implementadas na calculadora: soma, subtração, multiplicação, divisão, resto da divisão e o menos unário. Especificamente, você deve escrever uma biblioteca que implementa o tipo abstrato de dados **Polinômios**. A interface da sua biblioteca deve ser o arquivo `polinomios.h`, transcrito aqui:

```

/* ***** */
/*          Interface para o EP2: polinomios.h          */
/* ***** */
/*          Nao faca nenhuma alteracao neste arquivo.          */
/* ***** */

typedef void *polinomio;

polinomio cria();

polinomio leia();

```

```

polinomio copia(polinomio);
void      impr(char, polinomio);
polinomio soma(polinomio, polinomio);
polinomio subt(polinomio, polinomio);
polinomio nega(polinomio);
polinomio mult(polinomio, polinomio);
polinomio quoc(polinomio, polinomio);
polinomio rest(polinomio, polinomio);
void      libera(polinomio);

```

Note que o arquivo `polinomios.h` especifica o que se pode fazer com um polinômio (isto é, as operações sobre polinômios), mas não informa como um polinômio está armazenado. O tipo `polinomio` é definido como um apontador para `void`, que é essencialmente um apontador para um tipo arbitrário.

Abaixo descrevemos o que deve fazer cada uma das funções da biblioteca **Polinomios**:

- `polinomio cria();`

Devolve um polinômio nulo.

- `polinomio copia(polinomio p);`

Devolve uma cópia do polinômio `p`.

- `polinomio leia();`

Lê uma sequência de pares de números, seguida por um zero. Cada par consiste de um número do tipo `float`, representando um coeficiente, e um inteiro não negativo, representando um expoente. Devolve um novo polinômio com os termos dados pelos pares.

Exemplo: Se a sequência dada for

2 37 -7 0 5 11 -1 53 0

sua função deve devolver o polinômio $-x^{53} + 2x^{37} + 5x^{11} - 7$, armazenado numa lista encadeada no formato que você escolheu.

- `void impr(char c, polinomio p);`

Recebe um caracter que indica o nome do polinômio e um polinômio `p` e imprime o nome e `p` num formato adequado.

- `polinomio soma(polinomio p, polinomio q);`

Recebe dois polinômios, `p` e `q`, e devolve o polinômio que é a soma de `p` e `q`, ou seja, $p+q$.

Exemplo: Se $p(x) = -x^{53} + 2x^{37} + 5x^{11} - 7$ e $q(x) = 2x^{50} - 5x^{11} + x^2 - x$, sua função deve devolver o polinômio $-x^{53} + 2x^{50} + 2x^{37} + x^2 - x - 7$.

- `polinomio subt(polinomio p, polinomio q);`

Recebe dois polinômios, `p` e `q`, e devolve o polinômio que é a diferença de `p` e `q`, ou seja, $p-q$.

Exemplo: Se $p(x) = -x^{53} + 2x^{37} + 5x^{11} - 7$ e $q(x) = 2x^{50} - 5x^{11} + x^2 - x - 7$, sua função deve devolver o polinômio $-x^{53} - 2x^{50} + 2x^{37} + 10x^{11} - x^2 + x$.

- `polinomio nega(polinomio p);`

Recebe um polinômio `p`, e devolve o polinômio $-p$.

- `polinomio mult(polinomio p, polinomio q);`

Recebe dois polinômios, p e q , e devolve o polinômio que é o produto de p e q , ou seja, $p \cdot q$.

- `polinomio quoc(polinomio p, polinomio q);`

Recebe dois polinômios, p e q , e devolve o polinômio que é o quociente da divisão de p por q .

Exemplo: Se $p(x) = -8x^5 + 2x^3 + 5x^2 - 7$ e $q(x) = 2x^3 - x^2 + 2$, sua função deve devolver o polinômio $t(x) = -4x^2 - 2x$.

- `polinomio rest(polinomio p, polinomio q);`

Recebe dois polinômios, p e q , e devolve o polinômio que é o resto da divisão de p por q .

Exemplo: Se, como no exemplo anterior, $p(x) = -8x^5 + 2x^3 + 5x^2 - 7$ e $q(x) = 2x^3 - x^2 + 2$, sua função deve devolver o polinômio $r(x) = 13x^2 + 4x - 7$. De fato, o polinômio $r(x)$ tem grau estritamente menor que o polinômio $q(x)$ e você pode verificar que $p(x) = q(x) \cdot t(x) + r(x)$, para o $t(x)$ do exemplo anterior.

- `void libera(polinomio p);`

Libera a memória ocupada pelo polinômio p .

Nenhuma das funções acima modifica algum polinômio recebido como parâmetro. Todos os polinômios devolvidos por essas funções são alocados dinamicamente e o cliente da biblioteca deve liberá-los mediante chamadas à função `libera` quando não tiverem mais utilidade. Veja o cliente de teste disponibilizado.

A sua missão no EP2 é implementar a biblioteca `polinomios.c`, que corresponde à interface `polinomios.h`.

Como a biblioteca `polinomios.c` vai fazer leituras e impressões, porque vai conter a implementação das funções `leia` e `impr`, e vai alocar e desalocar memória, ela deve começar com os seguintes includes:

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

Dentro desta biblioteca, deve estar declarado o tipo das células das listas encadeadas que serão usadas para armazenar os polinômios. Note que os programas que forem usar a sua biblioteca de polinômios (os clientes da sua biblioteca) não precisam saber nada disso. Esses programas não precisam conhecer a estrutura que você vai utilizar, e nem sequer precisam saber que sua biblioteca vai usar listas encadeadas para armazenar os polinômios. Para escrever um cliente da sua biblioteca, é suficiente conhecer a interface da biblioteca (o arquivo `polinomio.h`) e saber o que fazem as funções dessa interface. Veja o cliente disponibilizado para a fase de testes da sua biblioteca.

Sua biblioteca pode conter funções auxiliares, não declaradas no arquivo `polinomios.h`. Estas são funções privadas da sua biblioteca, que o cliente não deve usar.

Requisitos adicionais sobre a implementação da biblioteca

- A implementação da sua biblioteca deve estar armazenada no arquivo `polinomios.c`.
- Sua implementação das funções `soma` e `subt` deve ser eficiente: os polinômios de entrada devem ser percorridos apenas uma vez.
- Sua implementação das funções `impr`, `nega` e `libere` deve ser eficiente: o polinômio de entrada deve ser percorrido apenas uma vez.
- Sua implementação das funções `mult`, `quoc` e `rest` também devem ser tão eficientes quanto possível.
- Sua biblioteca deve poder ser compilada separadamente do arquivo que contém o programa cliente da biblioteca. Veja o arquivo `Makefile` disponibilizado.

Programa cliente para fazer testes com a sua biblioteca

Para este EP, disponibilizamos um programa cliente que você deve usar para testar a implementação da sua biblioteca **Polinomios**. Esse arquivo será usado para testar a sua implementação, assim evite alterá-lo.

Disponibilizamos também um Makefile que você pode usar para compilar a sua biblioteca e o programa cliente. Ajuste o diretório que aparece no Makefile caso seja necessário.

Arquivos disponibilizados

- [polinomios.h](#)
- [polinomios.c](#) - esqueleto
- [testaEP2.c](#)
- [Makefile](#)

Exemplo de saída do programa

```
*****
Calculadora de polinomios - fase de testes
*****
```

```
0: cria e mostra um polinomio nulo
l: le e mostra um polinomio
c: le, copia e mostra a copia de um polinomio
+: le dois polinomios e mostra sua soma
-: le dois polinomios e mostra sua diferenca
*: le dois polinomios e mostra seu produto
/: le dois polinomios e mostra o quociente da sua divisao
%: le dois polinomios e mostra o resto da sua divisao
n: le um polinomio e mostra o seu negativo
q: sair do programa
```

Digite a opcao desejada: 0

$p(x) = 0$

```
0: cria e mostra um polinomio nulo
l: le e mostra um polinomio
c: le, copia e mostra a copia de um polinomio
+: le dois polinomios e mostra sua soma
-: le dois polinomios e mostra sua diferenca
*: le dois polinomios e mostra seu produto
/: le dois polinomios e mostra o quociente da sua divisao
%: le dois polinomios e mostra o resto da sua divisao
n: le um polinomio e mostra o seu negativo
q: sair do programa
```

Digite a opcao desejada: l

Digite o coeficiente e o expoente de cada termo de um polinomio, seguido de um 0.
1 2 3 4 0

$p(x) = 1.00 x^2 + 3.00 x^4$

```
0: cria e mostra um polinomio nulo
l: le e mostra um polinomio
c: le, copia e mostra a copia de um polinomio
+: le dois polinomios e mostra sua soma
-: le dois polinomios e mostra sua diferenca
*: le dois polinomios e mostra seu produto
/: le dois polinomios e mostra o quociente da sua divisao
%: le dois polinomios e mostra o resto da sua divisao
n: le um polinomio e mostra o seu negativo
q: sair do programa
```

Digite a opcao desejada: c

Digite o coeficiente e o expoente de cada termo de um polinomio, seguido de um 0.

1 2 3 4 0

$$p(x) = 1.00 x^2 + 3.00 x^4$$

$$r(x) = 1.00 x^2 + 3.00 x^4$$

0: cria e mostra um polinomio nulo
 l: le e mostra um polinomio
 c: le, copia e mostra a copia de um polinomio
 +: le dois polinomios e mostra sua soma
 -: le dois polinomios e mostra sua diferenca
 *: le dois polinomios e mostra seu produto
 /: le dois polinomios e mostra o quociente da sua divisao
 %: le dois polinomios e mostra o resto da sua divisao
 n: le um polinomio e mostra o seu negativo
 q: sair do programa

Digite a opcao desejada: +

Digite o coeficiente e o expoente de cada termo do primeiro polinomio, seguido de um 0.
 1 2 3 4 0

Digite o coeficiente e o expoente de cada termo do segundo polinomio, seguido de um 0.
 2 1 1 4 -1 6 0

$$p(x) = 1.00 x^2 + 3.00 x^4$$

$$q(x) = 2.00 x^1 + 1.00 x^4 + -1.00 x^6$$

$$r(x) = 2.00 x^1 + 1.00 x^2 + 4.00 x^4 + -1.00 x^6$$

0: cria e mostra um polinomio nulo
 l: le e mostra um polinomio
 c: le, copia e mostra a copia de um polinomio
 +: le dois polinomios e mostra sua soma
 -: le dois polinomios e mostra sua diferenca
 *: le dois polinomios e mostra seu produto
 /: le dois polinomios e mostra o quociente da sua divisao
 %: le dois polinomios e mostra o resto da sua divisao
 n: le um polinomio e mostra o seu negativo
 q: sair do programa

Digite a opcao desejada: -

Digite o coeficiente e o expoente de cada termo do primeiro polinomio, seguido de um 0.
 1 1 2 2 0

Digite o coeficiente e o expoente de cada termo do segundo polinomio, seguido de um 0.
 2 2 3 3 0

$$p(x) = 1.00 x^1 + 2.00 x^2$$

$$q(x) = 2.00 x^2 + 3.00 x^3$$

$$r(x) = 1.00 x^1 + -3.00 x^3$$

0: cria e mostra um polinomio nulo
 l: le e mostra um polinomio
 c: le, copia e mostra a copia de um polinomio
 +: le dois polinomios e mostra sua soma
 -: le dois polinomios e mostra sua diferenca
 *: le dois polinomios e mostra seu produto
 /: le dois polinomios e mostra o quociente da sua divisao
 %: le dois polinomios e mostra o resto da sua divisao
 n: le um polinomio e mostra o seu negativo
 q: sair do programa

Digite a opcao desejada: n

Digite o coeficiente e o expoente de cada termo de um polinomio, seguido de um 0.
 1 2 3 4 0

$$p(x) = 1.00 x^2 + 3.00 x^4$$

$$r(x) = -1.00 x^2 + -3.00 x^4$$

0: cria e mostra um polinomio nulo
l: le e mostra um polinomio
c: le, copia e mostra a copia de um polinomio
+: le dois polinomios e mostra sua soma
-: le dois polinomios e mostra sua diferenca
*: le dois polinomios e mostra seu produto
/: le dois polinomios e mostra o quociente da sua divisao
%: le dois polinomios e mostra o resto da sua divisao
n: le um polinomio e mostra o seu negativo
q: sair do programa

Digite a opcao desejada: q
Tchau!

Recomendações

Utilize listas encadeadas no seu programa para armazenar cada um dos polinômios. Se você não utilizar listas encadeadas, o seu EP receberá nota zero.

Faça a sua biblioteca aos poucos.

Primeiro implemente as funções [cria](#), [impr](#), [leia](#), [libere](#). Teste estas funções e só prossiga quando elas estiverem funcionando corretamente.

Depois implemente a função [copia](#). Teste até ela estar correta.

Depois implemente a função [nega](#). Teste até ela estar correta.

Passa para a função [soma](#). Teste até ela estar correta.

Passa para a função [subt](#). Teste até ela estar correta.

Finalmente prossiga da mesma maneira para as funções [mult](#), [quoc](#) e [rest](#).

Siga as recomendações da página com [informações sobre os EPs](#).

Divirta-se com o EP!
