

- Prazo para submissão: 05 Maio 2023
- Peso da atividade: 1

Listas Ligadas

Contextualização

A atividade consistirá de 3 atividades simples

1. Derivar polinômio
2. Calcular $P(x)$ dado x
3. Tamanho da lista em tempo $\mathcal{O}(1)$

Atividade 1 - Peso 0.3

Nome do arquivo para submissão: ListasLigadas_a.c

Você deve implementar uma função que recebe um polinômio e retorna sua derivada. Sendo a leitura do polinômio inicial e da derivada realizadas através de listas ligadas.

O *input* consistirá de diversas linhas. A primeira representa o número de termos, seguida por linhas contendo dois números representando o coeficiente e o expoente de um termo polinomial, onde o coeficiente e o expoente são inteiros. O input será passado como um arquivo txt para a main (*Dica: O nome do arquivo pode ser acessado em `argv[1]`*).

Ex:

```
5
10 2
5 5
8 3
-3 13
-8 0
```

Representa o polinômio $P(x) = -3x^{13} + 5x^5 + 8x^3 + 10x^2 - 8$.

Sendo sua derivada $P'(x) = -39x^{12} + 25x^4 + 24x^2 + 20x^1$

O *output* deve ser:

```
- 39.00x^12 + 25.00x^4 + 24.00x^2 + 20.00x^1
```

O output deve ter 2 casas decimais para cada coef.

(Dica: A função *ImprimePolinomio* passada pelo professor já imprime nesse formato.)

Atividade 2 - Peso 0.3

Nome do arquivo para submissão: ListasLigadas_b.c

Você deverá implementar uma função para calcular o valor do polinômio em um dado valor x .

O *input* será equivalente ao input da atividade 1, com o valor de x sendo passado como o segundo argumento para a função *main* (Dica: Você terá acesso a string contendo esse número através do termo *argv[2]*, onde *argv[1]* continua sendo o nome do arquivo contendo o polinômio.)

Ex:

```
5
10 2
5 5
8 3
-3 13
-8 0
4
```

E o *output* deve ser apenas uma linha contendo o valor do polinômio para o x dado. Neste exemplo, teríamos:

$$P(4) = 10 \cdot 4^2 + 5 \cdot 4^5 + 8 \cdot 4^3 - 3 \cdot 4^{13} - 8 = -201320808$$

Output:

-201320808.000000

O *output* deve ter 6 casas decimais.

(Dica: 1 - Guardem a resposta como *double* para aumentar a precisão; 2 - O *%lf* para *double* já imprime com 6 casas decimais.)

Atividade 3 - Peso 0.4

Nome do arquivo para submissão: ListasLigadas_c.c

Você deve criar uma implementação de listas ligadas simples para polinômios com uma função de inserção de novos termos no polinômio, assim como uma função "int Tamanho(*L)" que recebe um *pointer* para a estrutura da sua lista e que retorna o tamanho da lista em tempo constante $\mathcal{O}(1)$, isto é, independente do tamanho da lista.

Qualquer teste que exista nessa etapa será de forma simbólica, e o seu código será analisado de forma individual para checar se sua implementação da estrutura de listas ligadas simples e da função, operam em tempo constante.

Instruções

- As funções para listas ligadas simples e polinômios apresentadas em aula podem ser livremente copiadas e alteradas.
- O objetivo é usar listas ligadas, então não adianta armazenar os polinômios de outra forma, correndo o risco de zerar a questão. Dito isto, podem usar a implementação já passada pelo professor, ou criar a sua própria, contanto que siga a ideia de uma lista ligada onde cada elemento é composto de algumas informações mais um apontador pro próximo elemento. Os coeficientes e expoentes dos polinômios podem ser valores racionais.
- Uma operação em tempo constante $\mathcal{O}(1)$ significa que independente do valores passados a ela, ela leva aproximadamente o mesmo tempo para ser realizada. Um exemplo é o acesso de um valor qualquer em uma array tradicional em C. Se temos uma array A de tamanho n e queremos acessar o $k + 1$ -ésimo valor ($k + 1 \leq n$), o computador irá realizar uma operação de soma para determinar o endereço da variável $\&A[k]$ ou $A + k$ e checar o valor presente neste endereço de memória. Sendo o tempo de adição entre dois números aproximadamente constante. Independente do tamanho da array passada, o tempo para acessar o valor pedido é constante, já que apenas uma operação de adição precisa ser realizada. Pensem no tempo de um algoritmo como a quantidade de operações que ele precisa realizar dado um certo *input*. Por exemplo um algoritmo que busca um certo valor em uma lista não ordenada de tamanho n . No pior dos casos ele vai ter que checar cada um dos elementos, comparando cada um com o valor procurado, até chegar no último, realizando um total de n comparações. Sendo assim possui complexidade $\mathcal{O}(n)$.