

Aluno: João Pedro Gubertt

Data: 12/04/2023

Disciplina: Estrutura de dados



## Trabalho de programação B (Documentação)

Primeiramente, uma captura de tela listando TODAS as funções do programa, incluindo TAD's e Funções próprias do programa:

```
//Inicializa a lista setando os valores de lista.comeco e lista.fim
void inicializarLDE(LDE& lista);
// percorre a lista ao contrário para deixar o que for de maior grau pela frente
void mostrarLDE(LDE lista, string frase);
//Insere normalmente na LDE, só que insere dois insere dois índices, para a constante e outro para o expoente
bool inserirLDE(LDE& lista, float constante, int expoente);
//Retorna um nó onde o índice de expoente do parâmetro seja igual
No* buscarLDE(LDE lista, int expoente);
//retira o item da lista onde o expoente seja o desejado;
bool retirarLDE(LDE& lista, int expoente);
//Função simples de soma para utilizar nos índices K
float soma(float k1, float k2);
//Função simples de subtração para utilizar nos índices K
float sub(float k1, float k2);
// Função que executará o menu, dando opções ao usuário
void menu();
//Multiplica os índices K por uma constante !IMPORTANTE, ALTERA A PRÓPRIA LISTA.
void multiplicarPorEscalar(LDE& lista, int x);
//Lê apenas uma lista de polinômios para escalar e valor numérico
void lerPolinomioUmaLista(LDE& lista);
//Lê duas listas para qualquer outro tipo de operação - soma, sub e multiplicação de polinômios
void lerPolinomioDuasListas(LDE& lista, LDE& lista1);
//Soma o índice K e mantém o E
LDE somaExp(LDE lista);
//Subtrai o índice K e mantém o E
LDE subPolinomios(LDE lista1, LDE lista2);
//SOMA DUAS LISTAS DE POLINOMIO
LDE somaPolinomios(LDE lista1, LDE lista2);
//multiplica um polinômio por outro
LDE produtoPolinomio(LDE lista1, LDE lista2);
//substitui o valor de x, potencia esse valor pelo índice e para depois multiplicar por E
float valorNumerico(LDE lista, float x);
```

## TAD de lista encadeada

Acima de cada protótipo de função está uma breve explicação, até a função retirar temos as funções nativas da LDE que o professor nos apresentou, com algumas mudanças. Em “mostrarLDE”, a mudança feita consiste em começar a imprimir todo o polinômio de trás para a frente, utilizando o auxiliar em “lista.fim”, pois normalmente polinômios são representados com o termo de maior grau no início, é aplicada uma lógica para mostrar também um polinômio corretamente, com os sinais, não mostrar o x em caso de expoente = 0, basicamente para deixar a parte graficamente mais parecida com polinômios. Em “inserirLDE”, o usuário entra com um índice K e um índice E, onde são respectivamente Constante e expoente da variável, e a forma de inserção (vazia, início, fim e no meio) agora é dado pelo valor do expoente, ou seja, de E. Da

mesma maneira que a “BuscarLDE” também utiliza o expoente para retornar algo em comum, que seria o nó de mesmo expoente na atualização dessa função nativa da LDE.

### Biblioteca de operações

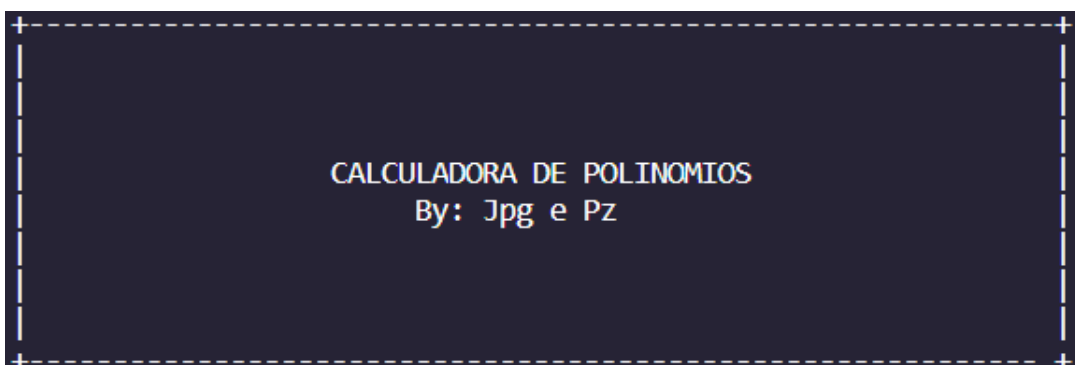
- Soma e Sub: Funções simples que somam e subtraem, são utilizadas em somaPolinomios e subPolinomios, onde são passados por parâmetro as constantes e executam o que precisam executar.
- somaExp: Uma função que retorna uma lista mais “limpa”, essa função tem como papel limpar possíveis repetições de expoentes para facilitar a escrita das funções posteriores que executarão soma, multiplicação e subtração. Basicamente percorre a lista até o final, utilizando um ponteiro para mudar o valor juntamente da função buscarLDE para achar expoentes iguais, o que irá retornar uma lista que não terá expoentes repetidos, o que posteriormente irá facilitar demais.
- somaPolinomios e subPolinomios: funções basicamente iguais onde dentro delas são utilizadas funções correspondente à suas operações. Primeiramente são inicializadas a lista resultado, e duas listas auxiliares: que serão fruto da função somaExp, com essas funções auxiliares iremos percorrer o laço principal enquanto nenhuma das duas listas tenham chego ao seu fim, A inserção na lista que será retornada como resultado dessas funções ocorrerá da seguinte maneira, no laço comentado, os expoentes das auxiliares que percorrem será comparado, se forem equivalentes a operação específica será feita, e o resultado será armazenado em uma variável do tipo float, se o resultado for diferente de 0, será inserido na lista que será retornada ao fim da execução da função e também ambos os nós auxiliares irão andar uma posição, caso não tenha acontecido essa hipótese de expoentes equivalentes o expoente do auxiliar de menor grau de expoente será inserido na lista que será retornada ao fim da função e esse auxiliar andar uma posição, isso acontece pois a ordem de expoentes está crescente na lista, e somente na exibição de tela a lista percorre ao contrário, ou seja, é possível que após eliminar esse de menor grau o próximo da lista possua o expoente igual ao de outro. Após esse laço abater a exigência é possível que tenham sobrado valores que não foram adicionados à lista, **em caso de listas de tamanhos diferentes**, isso é corrigido com laços que percorrem a auxiliar que não chegou a NULL inserindo diretamente na lista. É importante ressaltar que a sub seja o segundo parâmetro e zero como primeiro em casos onde não possua termo para subtrair, pois como a fórmula da subtração torna os termos do segundo polinômio todos negativos os números serão passados com o sinal oposto (Ex:  $0 - (+2x^2) = -2x^2$ ), o mesmo é feito na soma, mas como não ocorre essa troca de sinal é um passo desnecessário.
- lerPolinomiosUmaLista e lerPolinomiosDuasListas: Conforme a necessidade de cada operação é utilizada um tipo de leitura pelo usuário do polinômio ou dos polinômios, é o que essas funções abordam, onde são necessários duas listas, é feita a inserção com inserirLDE, exibindo a maneira correta de inserir cada polinômio, no caso de uma lista é feito somente uma entrada, então é questionado ao usuário somente se deseja inserir mais um monômio, quando se lê duas listas esse processo ocorre duas vezes, para lista um e dois, que são incrementadas via referência.

- **produtoPolinomio:** Uma função que percorre duas listas (simultaneamente) multiplicando o valor de K e somando o valor de E, utilizando também o somaExp para deixar todo o processo mais fácil, nessa função é retornada uma LDE. O uso dos dois laços garante que um termo multiplique todos e aí sim passe para o próximo termo a multiplicar.
- **valorNumerico:** Função que retorna um float que é basicamente a substituição da variável por um float, onde elevamos esse número que recebemos por parâmetro por seu grau e multiplicamos com a constante, acreditamos ser a lógica mais simples do trabalho, utilizamos a biblioteca math.h para usar a função pow() que é basicamente potenciação onde o primeiro parâmetro é o valor recebido na chamada da função valorNumérico e o segundo é o aux->e , ou seja, o expoente.
- **multiplicarPorEscalar:** Diferentemente da foto foi necessário efetuar uma mudança para retornar uma lista e obedecer um padrão, agora é gerada uma lista nova onde é inserido o resultado da multiplicação do parâmetro pelo K do item da lista que está no laço, para isso foi necessário criar uma auxiliar para utilizar inserirLDE e assim conseguir multiplicar sem erros, podendo ser multiplicado por números negativos.
- **menu:** É função que engloba a utilização da leitura de listas e a escolha do usuário na aplicação final, é utilizado um switch case, com um case para cada tipo de aplicação, onde cada uma, que necessita, pede sua particularidade, o valor numérico por exemplo, pede o valor de x, em multiplicação por escalar é pedido um float para multiplicar o polinômio, no restante a entrada utilizada são de duas listas e a operação ocorre conforme tem que acontecer. A parte gráfica é bem simples, são alguns cout's seguidos de cin's e comandos para limpar tela, a condição de parada é especificada na leitura de polinômio, no caso o o usuário teria de digitar 1 para continuar inserindo monômios, fora isso, é exibida uma apresentação e uma tela que lista as opções.

Uma ressalva é de que majoritariamente essas funções utilizam o laço com um auxiliar setado no começo e percorrem até o fim, em caso de não citações de método de laço na explicação esse é o padrão.

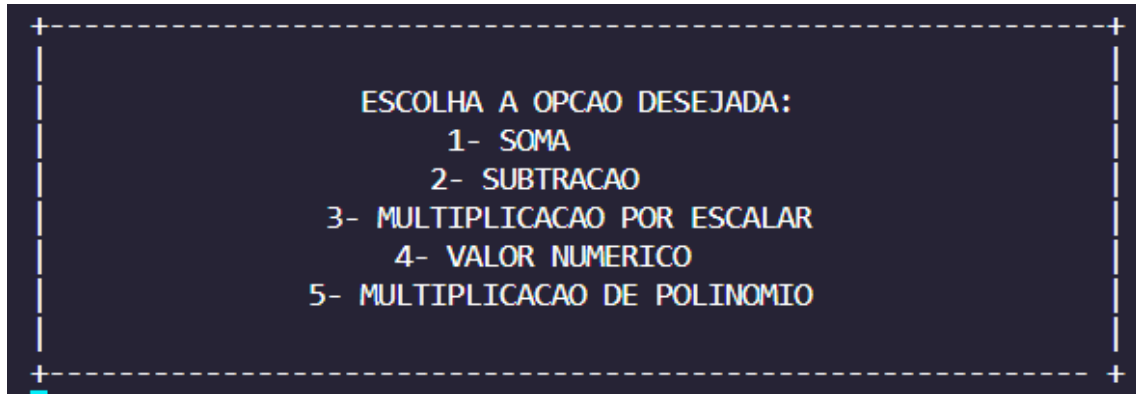
### Aplicação Final:

A calculadora de polinômios foi a aplicação escolhida, onde o usuário entra com seus termos e obtém a resposta, não há um mecanismo de memória que guarde a resposta anterior do usuário. Primeiramente é mostrado ao usuário uma tela, com um título:



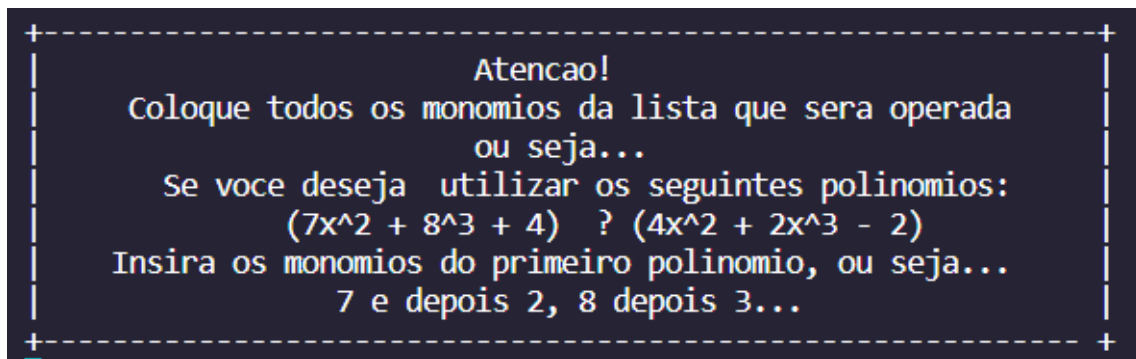
Mostrando os autores do trabalho, João Pedro Gubertt e Pedro Zunino (Que preferiu fazer esta parte sozinho).

Após um tempo é mostrado ao usuário suas opções com a calculadora, Agora esperando uma entrada com o teclado:

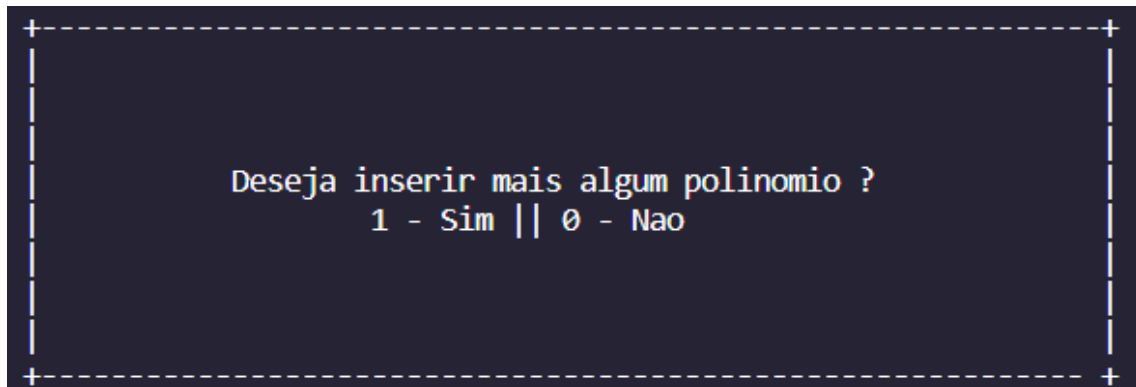


Irei escolher a opção soma e utilizar um exemplo do google para testar o resultado, polinômios escolhidos:  $(9x^2 - 13x - 5) + (16x + 10)$ , o resultado deverá ser  $9x^2 + 3x + 5$ .

Ao entrar com o 1 (soma), a mensagem seguinte é exibida:



Então entrei com 9, 2, e a próxima mensagem será:



Como queremos inserir mais digitamos um, caso não quisermos mais será 0, então vou inserir respectivamente -13, 1, 1(para inserir mais um), -5 e 0, por fim 0, então aparecerá na tela:

```

+-----+
|                                     |
|               Atencao!             |
|         Coloque todos os monomios da lista que sera operada         |
|                   ou seja...       |
|         Se voce deseja utilizar os seguintes polinomios:             |
|         (7x^2 + 8^3 + 4) ? (4x^2 + 2x^3 - 2)                         |
|         Insira os monomios do primeiro polinomio, ou seja...         |
|                   4 e depois 2, 2 depois 3...                         |
|                                     |
+-----+

```

Para a inserção do segundo polinômio, a sequência aqui será 16, 1, 1, 10, 0 e 0 para finalizar a entrada.

```

1o Polinomio soma: 9x^2 -13x -5
2o Polinomio soma: 16x + 10

Resultado Soma: 9x^2 + 3x + 5

```

Esse é o resultado, que permanece na tela por 5 segundos, sendo substituído por a próxima mensagem:

```

+-----+
|                                     |
|               DESEJA REALIZAR OUTRA CONTA?                             |
|               1- Sim || 2 - Nao                                         |
|                                     |
+-----+

```

Como vamos continuar o teste apertamos 1, então a tela de apresentação aparecerá novamente e escolhermos a próxima operação. Vamos testar a subtração agora.  $(2x^3 - 5x^2 + 21) - (2x^3 + x^2 + 2x + 5)$  que deverá resultar  $-6x^2 - 2x + 16$ . Na tela:

```

1o Polinomio sub: 2x^3 -5x^2 + 21
2o Polinomio sub: 2x^3 + 1x^2 + 2x + 5

Resultado Sub: -6x^2 -2x + 16

```

Teste Multiplicação por escalar, exemplo do PDF do trabalho,  $(3x^2 - 5x + 8) \cdot (-2)$ , o resultado é  $-6x^2 + 10x - 16$ . Segue feito pelo programa:

```

Polinomio a ser multiplicado: 3x^2 -5x + 8
Escalar que ira multiplicar: -2
Resultado multiplicacao por escalar: -6x^2 + 10x -8

```

Valor numérico segundo o exemplo:  $x^3 - 2x^2 + 4x$ , quando  $x = 2$ ,  $2^3 - 2.2^2 + 4.2 = 8 - 8 + 8 = 8$ ,  
segundo o programa:

ENTRE COM UM VALOR PARA A  
A VARIÁVEL X

Mensagem do caso.

Atencao!!

Insira todos os monomios  
Primeiro a sua constante e depois seu expoente  
independente da ordem e sinal!

Ainda não foi citado, porém essa é a mensagem de leituras de polinômios únicos, utilizada em multiplicação por escalar e valor numérico.

```
Polinomio recebido: 1x^3 -2x^2 + 4x
Valor de X2
Valor numerico do polinomio eh igual a 8
```

Por fim produto de polinômio, exemplo  $f(x) = x + 2x^2 + 3x^3$  por  $g(x) = 4 + 5x + 6x^2$ , resulta em  $18x^5 + 27x^4 + 28x^3 + 13x^2 + 4x$ . Pelo programa:

```
1o polinomio para multiplicacao: 3x^3 + 2x^2 + 1x
2o polinomio para multiplicacao: 6x^2 + 5x + 4
Resultado produto dos polinomios: 18x^5 + 27x^4 + 28x^3 + 13x^2 + 4x
```

Testes realizados.

Aplicação Funcionando.