

PROGRAMAÇÃO AVANÇADA – 1ª avaliação
PROFESSOR: ADELARDO ADELINO DANTAS DE MEDEIROS

ALUNO: _____

MATRÍCULA: _____

Escreva uma classe em C++, denominada `Poly`, capaz de representar polinômios $P(x)$ de qualquer grau n ($n \geq 0$) com coeficientes reais, do tipo:

$$P(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1} + a_nx^n$$

Um polinômio de grau n será armazenado e representado internamente pelos seus $n+1$ coeficientes (ou seja, um array de $n+1$ números reais).

Utilizando essa classe, desenvolva as seguintes funcionalidades:

- Crie os construtores (*default*, por cópia e específico) e o destrutor apropriados.
 - O construtor *default* deve criar um polinômio de grau 0 cujo valor do único coeficiente, a_0 , será 0.0.
 - Um construtor específico deve criar um polinômio do grau passado como parâmetro. Se o grau for nulo, o polinômio deve ser criado tal como no construtor *default*. Se o grau for maior que zero, o polinômio deve ter todos os coeficientes nulos, com exceção do último, a_n , que terá valor 1.0. Verifique se o construtor específico pode ou não ser utilizado como conversor de tipo (ou seja, se ele deve ou não ser *explicit*).
- Sobrecarregue o operador de atribuição (`operator=`). Lembre-se que o objeto que está sendo atribuído poderá já conter informação anterior, sendo necessário prever a liberação de memória que eventualmente já tenha sido alocada previamente.
- Defina um método (função-membro) `getValor` ou sobrecarregue o `operator()` para retornar o valor do polinômio para um dado valor real de x , que é passado como parâmetro.
- Sobrecarregue o operador `<<` para escrever na tela um polinômio qualquer. Não há necessidade de aprimorar o aspecto da impressão, que pode ser na forma do exemplo a seguir ou de qualquer outra forma que você escolher, desde que os valores de todos os coeficientes sejam exibidos. Também não há necessidade de lidar com situações especiais (não imprimir o termo correspondente quando um coeficiente for nulo; substituir `+` por `-` quando o coeficiente for negativo, etc.)

$$5.7 + 0.0*x + -3.2*x^2 + -0.2*x^3$$

- Sobrecarregue o operador `>>` de tal forma que permita que o usuário digite o grau e os coeficientes de um `Poly` qualquer.
 - Lembre-se que o objeto poderá já conter informação anterior, sendo necessário prever a liberação de memória que eventualmente já tenha sido alocada previamente.
 - Não se esqueça de fazer a crítica dos dados digitados, pois:
 - a) Um polinômio não pode ter grau negativo (mas pode ter grau zero); e
 - b) O último coeficiente não pode ser nulo, exceto no caso do polinômio de grau 0, pois então o grau seria menor que o valor digitado.
- Sobrecarregue o operador `+` para fazer a soma de dois polinômios, retornando o resultado. Lembre-se que, em alguns casos, a soma pode gerar polinômios de grau inferior ao esperado. Por exemplo, a soma de dois polinômios de segundo grau pode gerar um polinômio de primeiro grau, caso o último coeficiente do polinômio resultante seja nulo:

$$(6 + 5x - 3x^2) + (-1 + x + 3x^2) = 5 + 6x$$

- Sobrecarregue o operador `*` para retornar o produto de dois polinômios.

Usando a classe `Poly`, faça um programa principal que execute as seguintes operações:

1. Solicite ao usuário que digite o grau e os coeficientes de dois polinômios, $Q_1(x)$ e $Q_2(x)$.
2. Imprima os polinômios $Q_1(x)$ e $Q_2(x)$.
3. Calcule a soma $S = Q_1(x) + Q_2(x)$.
4. Imprima:
 - O polinômio S .
 - O valor da soma para $x = -3.0$, ou seja, $S(-3.0)$ ou $S.getValor(-3.0)$.
5. Calcule o produto $P = Q_1(x) \times Q_2(x)$.
6. Imprima:
 - O polinômio P .
 - O valor do produto para $x = 1.0$, ou seja, $P(1.0)$ ou $P.getValor(1.0)$.