Algoritmos e Estruturas de Dados II

Lista 1: Tipos Abstratos de Dados

Nome:_____

Matrícula:_____

Introdução: Um polinômio de grau **n** é uma função do tipo $P(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$, onde a_0, a_1, a_2, ... , a_n pertencem ao conjunto dos números reais e $a_n \neq 0$. Qualquer polinômio de grau **n** pode ser representado em um programa de computador por um vetor **p** com **n + 1** posições, onde cada posição **p[i]** do vetor armazena o valor do coeficiente **ai**, $i \in \{0, ..., n\}$. Por exemplo: o polinômio de grau **4**, $Q(x) = 5 + 3x^2 + 2x^4 = 5x^0 + 0x^1 + 3x^2 + 0x^3 + 2x^4$, pode ser representado pelo vetor "float q[**5**] = {5, 0, 3, 0, 2};"

Questão 1. Crie um TAD Polinômio com a seguinte interface:

```
class Polinomio {
// Dados privados.
private:
  int n;
  float a[GRAU_MAXIMO + 1];

// Métodos públicos.
public:
  // Cria um polinômio igual a P(x)=0.
  Polinomio();

  // Cria um polinômio a partir de um vetor q com m elementos.
  Polinomio(int m, float* q);

  // Retorna o grau do polinômio.
  int grau();

  // Retorna o coeficiente a[i].
  float get(int i);

  // Atribui o valor 'b' ao coeficiente a[i].
  void set(int i, float b);
```

```
  // Retorna o valor do polinômio corrente no ponto x.
  float Avaliar(float x);

  // Faz com que o polinômio corrente fique igual ao polinômio q
  // passado como parâmetro.
  void Atribuir(Polinomio& q);

  // Atribui ao polinômio corrente a soma dos polinômios p1 e p2
  // de mesmo grau passados como parâmetro.
  void Somar(Polinomio& p1, Polinomio &p2);

  // Faz com que o polinômio corrente fique igual a derivada do polinômio q
  // passado como parâmetro.
  void Derivar(Polinomio& q);

  // Faz com que o polinômio corrente fique igual a integral do polinômio q
  // passado como parâmetro.
  void Integrar(Polinomio& q);
};
```

Questão 2. Implemente o método **float Polinomio::Avaliar(float x)** que retorna o valor do polinômio corrente no ponto **x**.

Questão 3. Implemente o método **void Polinomio::Atribuir(Polinomio& q)** que f**az com que o polinômio corrente fique igual ao polinômio q** passado como parâmetro.

Questão 4. Implemente o método **void Polinomio::Somar(Polinomio& p1, Polinomio &p2)** que atribui ao polinômio corrente a soma dos polinômios **p1** e **p2** de mesmo grau passados como parâmetro.

Questão 5. Implemente o método **void Polinomio::Derivar(Polinomio& q)** que faz com que o polinômio corrente fique igual a derivada do polinômio **q** passado como parâmetro.

DICA: A derivada de um polinômio de grau **n** > 1, $P(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$, é um polinômio de grau **n - 1** e pode ser calculada da seguinte forma: $P'(x) = a_1 + 2a_2x + 3a_3x^2 + ... + na_nx^{n-1}$. Por exemplo: $Q'(x) = 2*3*x^{(2-1)} + 4*2*x^{(4-1)} = 6x + 8x^3$.

Questão 6. Implemente o método **void Polinomio::Integrar(Polinomio& q)** que faz com que o polinômio corrente fique igual a integral do polinômio **q** passado como parâmetro.

DICA: A integral de um polinômio de grau **n**, $P(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$, é um polinômio de grau **n + 1** e pode ser calculada da seguinte forma: *integral*$[P(x)] = c + a_0x + {}^{a1}/_{2}x^2 + {}^{a2}/_{3}x^3 + ... + {}^{a}{}_{n}/_{(n+1)}x^{n+1}$. Assuma que a constante **c** é igual a 0. Por exemplo: *integral*$[Q(x)] = 5x + {}^{3}/_{3}x^3 + {}^{2}/_{5}x^5 = 5x + x^3 + 0.4x^5$.