[1,0 PONTO] Escreva um programa que leia o **valor** de uma dívida, uma **taxa de juros** fixa e o número de **prestações** para quitação e apresente o **valor total a ser pago** considerando que as prestações começam a ser cobradas dentro de 1 mês. Por exemplo, seja uma dívida de 1000 reais a 5% de juros para pagar em 12 prestações. Teremos:

**Mês Dívida total Prestação Dívida restante**

0 1000,00 - 1000,00

1 1000,00 \* 1.05 = 1050,00 1050,00/12 = 87,5 1050-87,5 = 962,5

2 962,5\*1,05 = 1010,625 1010,625/11 = 91,875 ...

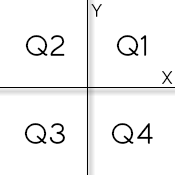
... ... ... ...

12 ? ? 0,0

1. [2,0 PONTOS] Escreva um programa que leia um dado **n positivo** e, em seguida, leia uma **sequência de n pares** devalores reais (x e y), que representam as coordenadas de pontos em um plano. A seguir, determine o quadrante ao qual pertence o ponto, ou se está sobre um dos eixos cartesianos ou na origem (x = y = 0).

Se o ponto estiver na origem, escreva a mensagem “Origem”.

Se o ponto estiver sobre um dos eixos escreva “Eixo X” ou “Eixo Y”, conforme for a situação.



1. [2,0 PONTOS] Considerando que a **série de Fibonacci** possui seus primeiros termos F0 = 0 e F1 = 1, os demais termos devem ser calculados a partir da soma dos dois anteriores: **Fn = Fn-1+ Fn-2**. Elabore um programa Java que leia um dado n positivo e apresente o enésimo termo da sequência Fibonacci usando vetores (array).
2. [5,0] Crie a classe ***IntegerSet*** para representar um conjunto cujos elementos são números inteiros no intervalo de 0 a 100. Tal representação deve ser implementada sobre um array de *booleans*, ou seja, o elemento do array **itens[i]** é *true* se o inteiro **i** estiver no conjunto. O elemento do array **itens[j]** é *false* se o inteiro **j** não estiver no conjunto. Por exemplo:

0 1 2 3 4 ... 100

A = { 1, 2, 3, 100} A.itens[ ] false **true** **true true** false ... **true**

O construtor sem argumentos deve inicializar o array **itens[]** como “conjunto vazio” (isto é, com todos os valores do array iguais a *false*). Por exemplo: IntegerSet A = new IntegerSet();

0 1 2 3 4 ... 100

A = { } A.itens[ ] false false false false false ... false

Um segundo construtor deve receber como argumento um vetor de inteiros. Nesse caso o array **itens[]** deve ser inicializado com *true* nas respectivas posições recebidas por parâmetro.

int b[] = {1, 2, 3};

IntegerSet A = new IntegerSet(b);

0 1 2 3 4 ... 100

A = {1, 2, 3} A.itens[ ] false **true** **true true** false ... false

Além dos construtores, a classe deve fornecer os seguintes métodos:

1. insertElement: insere um novo inteiro **k** no conjunto instanciado. Por exemplo: **A.insertElement(4);**

0 1 2 3 4 ... 100

A = {1, 2, 3, **4**} A.itens[ ] false **true** **true true** **true** ... false

1. deleteElement: exclui o inteiro **m** do conjunto instanciado. Por exemplo: **A.deleteElement(2);**

0 1 2 3 4 ... 100

A = {1, 3, 4} A.itens[ ] false **true** false **true** **true** ... false

1. union: cria um terceiro conjunto representando a união dos elementos de dois conjuntos existentes (os itens desse conjunto serão configurados como *true* se esse elemento for *true* em qualquer um dos conjuntos existentes ou em ambos, caso contrário o elemento deve ser configurado como *false*);

Por exemplo: C = A.union(B);

0 1 2 3 4 ... 100

A = { 0, 3, 4} A.itens[ ] **true** false false **true** **true** ... false

0 1 2 3 4 ... 100

B = { 1, 3, 100} B.itens[ ] false **true** false **true** false ... **true**

0 1 2 3 4 ... 100

C = { 0, 1, 3, 4, 100} C.itens[ ] **true** **true** false **true** **true** ... **true**

1. intersection: cria um terceiro conjunto representando a intersecção dos elementos de dois conjuntos existentes (os itens desse conjunto serão configurados como *false* se esse elemento for *false* em qualquer um dos conjuntos existentes ou em ambos, caso contrário o elemento deve ser configurado como *true*);

Por exemplo: C = A.intersection (B);

0 1 2 3 4 ... 100

A = { 0, 3, 4} A.itens[ ] **true** false false **true** **true** ... false

0 1 2 3 4 ... 100

B = { 1, 3, 100} B.itens[ ] false **true** false **true** false ... **true**

0 1 2 3 4 ... 100

C = { 3} C.itens[ ] falsefalsefalse **true** false ... false

1. toSetString: retorna uma String contendo os elementos presentes no conjunto instanciado separados por espaços;

Elabore o **diagrama de classes** para sua classe e escreva um **programa** para testá-la usando vários objetos *IntegerSet* e realizando as diversas operações entre eles.

Boa prova!