pvh( );

Testes de Software

Grupo de Automação de Testes

IFG – Inhumas

profvictorhugo.esy.es

github.com/TDD-IFG

A presente lista de exercícios leva em consideração todos os conceitos acumulados até o momento, em especial, a automação de testes unitários com uso de ferramentas, como o JUnit.
Abaixo segue uma lista de programas, ou partes de programas, seguido da descrição dos objetivos do programa.

Os programas/classes se encontram no repositório github
https://github.com/TDD-IFG/Exercicios.git

---

01. A classe UltrapassaValor faz parte de um programa de contagem para mercado de capitais. Esta classe visa resolver um problema básico de contagem de valores de ações, útil para comparar ações de empresas, especialmente quando uma ultrapassa o patamar de outra. Por enquanto, a referida classe somente possui o método executa, que processa uma lista de inteiros, sendo o primeiro valor atribuído a X, e o maior valor da lista atribuído a Z (X <= Z). O objetivo é descobrir quantos números inteiros, em sequência, devem ser somados a X (incluindo X), para que X ultrapasse Z o mínimo possível.

A entrada pode conter, por exemplo, os valores 21 21 15 30. Neste caso, é então assumido o valor 21 para X enquanto os valores 21 e 15 devem ser desconsiderados pois são menores ou iguais a X. Como o valor 30 está dentro da especificação (maior do que X) ele será válido e então deve-se processar os cálculos para apresentar na saída o valor 2, pois é a quantidade de valores somados para se produzir um valor maior do que 30 (21 + 22).

Como saída, espera-se somente um inteiro, que representa a quantidade de números inteiros que devem ser somadas, de acordo com a especificação acima.

Exemplos de entradas:

3	1	20					saída → 5						
1	0	0	0	0	100		saída → 14						
10	1	2	5	8	11		saída → 2						
10	1	2	5	8	1	2	5	8	1	1	2	5	8
	1	1	2	5	8	14	saída → 2						

O objetivo do teste é garantir o funcionamento do método, e seu grau de acertividade.

02. Jogo Big Bang Theory

Em um famoso seriado, dois amigos jogam uma variação do jogo japonês Pedra-Papel-Tesoura, para a resolução de impasses. O jogo precisa ser rapidamente decidido, mas um dos participantes não aceita perder. As regras do jogo:

1.o papel embrulha a pedra;
2.a tesoura corta o papel;
3.a pedra esmaga o lagarto;
4.o lagarto envenena Spock;
5.Spock destrói a tesoura;
6.a tesoura decapita o lagarto;
7.o lagarto come o papel;
8.o papel contesta Spock;
9.Spock vaporiza a pedra;
10.a pedra quebra a tesoura.

A classe JogoPPTLS foi desenvolvida, com o método jogo, que recebe duas Strings, sendo as jogadas dos jogadores 1 e 2, respectivamente. Ele decide quem ganhou, e retorna uma String com uma frase, que indica o resultado. Se o primeiro jogador vence, retorna-se “Bazinga!”, ou “Raj trapaceou!”, caso contrário. Caso ocorra empate, retorna-se “De novo!”.

O objetivo do teste é garantir o funcionamento do método, e seu grau de assertividade.

03. A classe Heroi foi desenvolvida para auxiliar na descoberta de bandidos que já foram capturados por um determinado Herói. Após diversas batalhas, todos os vilões haviam sido capturados pelo Herói e a sensação de segurança parecia fazer parte dos cidadãos novamente. Toda essa tranquilidade atípica deixou de existir a dois dias, quando um vilão fugiu. A classe Heroi foi desenvolvida pelo departamento de polícia, para permitir a rápida identificação dos vilões já capturados.

A classe possui 2 métodos, sendo o primeiro que recebe um array de Strings, contendo nomes de vilões, retornando ‘Y’, caso o vilão tenha sido capturado, ou ‘N’, caso contrário.

O objetivo do teste é garantir o funcionamento do método, e seu grau de assertividade.

04. A classe PoligonoRegular define características de polígonos, segundo a geometria. Ela ainda está incompleta, mas já deve ser capaz de calcular o seu próprio perímetro, através do método perímetro, que retorna um inteiro. Para funcionar, o método precisa obter N e L, sendo N o número de lados do polígono, e L o comprimento de cada lado.

O objetivo do teste é garantir o funcionamento do método, e seu grau de

assertividade.

05. Um novo jogo chamado IPR está se tornando muito popular. Esse jogo surgiu quando alguns amigos estavam sem conexão com a Internet, sem celular, sem computador e bastante desocupados. O jogo está tão popular que irá acontecer um campeonato mundial de IPR e cada país do mundo irá escolher um representante para competir.

O jogo funciona da seguinte forma: dois jogadores participam, o jogador 1 escolhe entre par ou ímpar, então cada jogador escolhe um inteiro positivo, se a soma desses números for par e o jogador 1 tiver escolhido par então o jogador 1 ganha, se a soma for ímpar o jogador 2 ganha. Caso o jogador 1 tivesse escolhido ímpar ele ganharia se a soma fosse ímpar, caso a soma fosse par o jogador 2 ganharia. Nada de diferente de um jogo de par ou ímpar convencional, correto?

A diferença do jogo é que o jogador 1 pode roubar e assim assegurar sua vitória independentemente do resultado do jogo de ímpar ou par convencional, já o jogador 2 pode ou não acusar o jogador 1 de roubo. Com essas adições no jogo se o jogador 1 roubar e o jogador 2 acusar o roubo então o jogador 2 ganha, caso o jogador 2 não acuse o roubo e o jogador 1 roubar então o jogador 1 ganha, caso o jogador 2 acuse o roubo, mas o jogador 1 não tiver roubado então o jogador 1 ganha, se o jogador 1 não roubar e o jogador 2 não acusar o roubo o jogo segue como descrito anteriormente.

A entrada consite de uma única linha contendo 5 inteiros: p,j1,j2,r,a. ($0 \leq p, r, a \leq 1$ e $1 \leq j1, j2 \leq 100$).

p representa a escolha do jogador 1 (se p= 1 então o jogador 1 escolheu par, se p = 0 então o jogador 1 escolheu ímpar). Os valores j1, j2, representam respectivamente o número escolhido pelo jogador 1 e pelo jogador 2. r representa se o jogador 1 roubou (se r = 1 então o jogador 1 roubou, se r = 0 então o jogador 1 não roubou), arepresenta se o jogador 2 acusou o roubo (se a = 1 então o jogador 2 acusou o jogador 1 de roubo, se a = 0 então ele não acusou o jogador 1 de roubo).

A classe IPR está parcialmente desenvolvida, e possui, por enquanto, o método Calcula, que recebe os parâmetros de entrada, e retorna um inteiro, indicando 1 para o jogador 1 vencedor, ou 2, para o caso do jogador 2 ter vencido.

Exemplos de entrada:

1 4 5 0 0 saída → 2

1 4 5 1 0 saída → 1

1 4 5 1 1 saída → 2

O objetivo do teste é garantir o funcionamento do método, e seu grau de assertividade. Isso precisa ser definido para as duas versões disponíveis para a classe, de modo a permitir decisão pela versão mais correta.

06. A classe Fortaleza está sendo construída para um jogo de RPG. Até o momento, ela possui o método executa, que serve para calcular o aumento de experiência (XP) de um personagem do jogo. O método calcula o aumento da XP do personagem (M) X vezes.

Exemplos de entrada:

```
1 544768710   saída →  544768710

2 538533133   saída →  1077066266

44 12451255   saída →  547855220
```

O objetivo do teste é garantir o funcionamento do método, e seu grau de assertividade.

07. A classe FalhaMotor foi desenvolvida para um fabricante de motores automotivos, para ser utilizado na fase de testes de novos motores. Observa-se a curva de velocidade do motor, em que as quedas de velocidade podem indicar problemas.

Entrada

A entrada é um teste do motor e é dada em duas linhas. A primeira tem o número N de medidas de velocidade do motor ($1 < N \leq 100$). A segunda linha tem N inteiros: o número de RPM (rotações por minuto) R_i de cada medida ($0 \leq R_i \leq 10000$, para todo R_i, tal que $1 \leq i \leq N$). Uma medida é considerada uma queda se é menor que a medida anterior.

Saída

A saída é o índice da medida em que houve a primeira queda de velocidade no teste. Caso não aconteça uma queda de velocidade a saída deve ser o número zero.

O método executa calcula o índice da media em que houve a primeira queda de velocidade.

Exemplo de entradas:

```
3
1 4 2       saída →    3


5
100 199 199 198 0      saída →    4
```

O objetivo do teste é garantir o funcionamento do método, e seu grau de assertividade.

08. Números em ponto flutuante podem ser bastante extensos para mostrar. Nesses casos, é conveniente usar a notação científica.

A classe ScientificNotation foi desenvolvida para, dado um número em ponto flutuante, converter este número para a notação científica: sempre mostrando o sinal da mantissa; sempre mostre 4 casas decimais na mantissa; usando o caractere 'E' para separar a mantissa do expoente; sempre mostre o sinal do expoente; e mostre o expoente com pelo menos 2 dígitos.

Entrada

A entrada é um número em ponto flutuante de dupla precisão X (de acordo com o padrão IEEE 754-2008). Nunca haverá um número com mais de 110 caracteres nem com mais de 6 casas decimais.

Saída

A saída é o número X em uma única linha na notação científica detalhada acima.

Exemplos de entrada/saída:

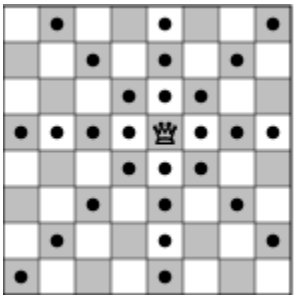
3.141592	+3.1416E+00

1.618033	+1.6180E+00

602214085774747474747474	+6.0221E+23

Ajude a definir qual das duas versões é mais correta. O objetivo do teste é garantir o funcionamento do método, e seu grau de assertividade, para cada uma das versões.

09. O jogo de xadrez possui várias peças com movimentos curiosos: uma delas é a *dama*, que pode se mover qualquer quantidade de casas na mesma linha, na mesma coluna, ou em uma das duas diagonais, conforme exemplifica a figura abaixo:



O grande mestre de xadrez Kary Gasparov inventou um novo tipo de problema de xadrez: dada a posição de uma dama em um tabuleiro de xadrez vazio (ou seja, um tabuleiro 8 × 8, com 64 casas), de

quantos movimentos, no mínimo, ela precisa para chegar em outra casa do tabuleiro?

Kary achou a solução para alguns desses problemas, mas teve dificuldade com outros, e por isso pediu o desenvolvimento de um programa que resolva esse tipo de problema.

Entrada

A entrada contém vários casos de teste. A primeira e única linha de cada caso de teste contém quatro inteiros X_1, Y_1, $X_2 \in Y_2$ ($1 \leq X_1, Y_1, X_2, Y_2 \leq 8$). A dama começa na casa de coordenadas (X_1, Y_1), e a casa de destino é a casa de coordenadas (X_2, Y_2). No tabuleiro, as colunas são numeradas da esquerda para a direita de 1 a 8 e as linhas de cima para baixo também de 1 a 8. As coordenadas de uma casa na linha X e coluna Y são (X, Y).

O final da entrada é indicado por uma linha contendo quatro zeros.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha na saída, contendo um número inteiro, indicando o menor número de movimentos necessários para a dama chegar em sua casa de destino.

Exemplo de Entrada	Exemplo de Saída
4 4 6 2 3 5 3 5 5 5 4 3 0 0 0 0	1 0 2

3 versões foram apresentadas pela empresa contratada. Seu papel é testar os executáveis e julgar a qualidade de cada um.

Ambiente recomendado: Eclipes/netbeans com java 8 e JUnit, em Ubuntu.

http://profvictorhugo.esy.es/programacao/java/como-usar-java-para-testar-funcionalidades-de-programas-em-c/

10. O programa SelecaoVetor1 lê um vetor A[100] float, e apresenta a posição e o valor, para os casos em que o valor armazenado é menor ou igual a 10.

Entrada

A entrada contém 100 valores, podendo ser inteiros, reais, positivos ou negativos.

Saída

Para cada valor do vetor menor ou igual a 10, escreva "A[i] = x", onde i é a posição do vetor e x é o valor armazenado na posição, com uma casa após o ponto decimal.

Exemplo de Entrada	Exemplo de Saída
0 -5 63 -8.5 ...	A[0] = 0.0 A[1] = -5.0 A[3] = -8.5 ...

O objetivo é testar o executável, assegurando sua funcionalidade e assertividade.

11. Mariazinha sabe que um Número Primo é aquele que pode ser dividido somente por 1 (um) e por ele mesmo. Por exemplo, o número 7 é primo, pois pode ser dividido apenas pelo número 1 e pelo número 7 sem que haja resto. Então ela pediu para você fazer um programa que aceite diversos valores e diga se cada um destes valores é primo ou não. Acontece que a paciência não é uma das virtudes de Mariazinha, portanto ela quer que a execução de todos os casos de teste que ela selecionar (instâncias) aconteçam no tempo máximo de um segundo, pois ela odeia esperar.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 200$), correspondente ao número de casos de teste. Seguem N linhas, cada uma contendo um valor inteiro X ($1 < X < 2^{31}$) que pode ser ou não, um número primo.

Saída

Para cada caso de teste imprima a mensagem “Prime” (Primo) ou “Not Prime” (Não Primo), de acordo com o exemplo abaixo.

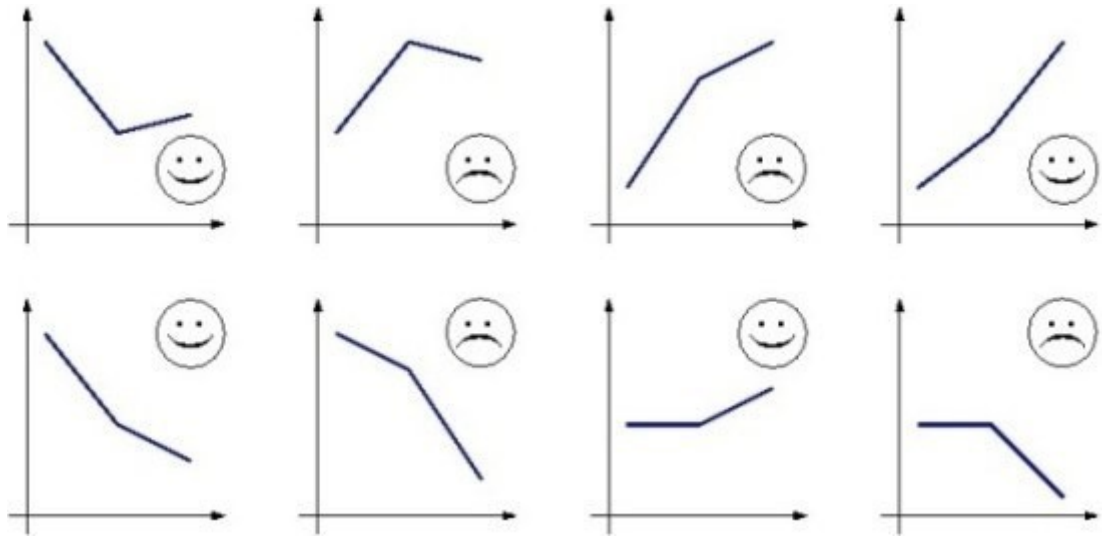
Exemplo de Entrada	Exemplo de Saída
3 123321 123 103	Not Prime Not Prime Prime

Para tal problema, o programa PrimoRapido foi desenvolvido. O cliente exigiu que o problema não ultrapasse 1 segundo, em qualquer caso. Desta forma, a empresa contratada entregou 2 versões, e precisamos descobrir qual a melhor, ou se há algum defeito.

12. Um *resort* de inverno brasileiro se preocupa muito com a satisfação dos seus hóspedes. Assim sendo, contratou empresa de pesquisas de comportamento, que mapeou os interesses dos hóspedes em buscar o *resort*, dependendo do clima, observando-se os últimos 3 dias. O mapeamento foi o seguinte:

Se a temperatura desceu do 1º para o 2º dia, mas subiu ou permaneceu constante do 2º para o 3º, as pessoas ficam felizes (primeira figura).

•Se a temperatura subiu do 1º para o 2º dia, mas desceu ou permaneceu constante do 2º para o 3º, as pessoas ficam tristes (segunda figura).
•Se a temperatura subiu do 1º para o 2º dia e do 2º para o 3º, mas subiu do 2º para o 3º menos do que subira do 1º para o 2º, as pessoas ficam tristes (terceira figura).
•Se a temperatura subiu do 1º para o 2º dia e do 2º para o 3º, mas subiu do 2º para o 3º no mínimo o tanto que subira do 1º para o 2º, as pessoas ficam felizes (quarta figura).
•Se a temperatura desceu do 1º para o 2º dia e do 2º para o 3º, mas desceu do 2º para o 3º menos do que descera do 1º para o 2º, as pessoas ficam felizes (quinta figura).
•Se a temperatura desceu do 1º para o 2º dia e do 2º para o 3º, mas desceu do 2º para o 3º no mínimo o tanto que descera do 1º para o 2º, as pessoas ficam tristes (sexta figura).
•Se a temperatura permaneceu constante do 1º para o 2º dia, as pessoas ficam felizes se subiu do 2º para o 3º dia ou tristes caso contrário (respectivamente, sétima e oitava figuras).



Entrada

A entrada consiste apenas de três inteiros, A, B e C (-100 ≤ A, B, C ≤ 100), os quais representam respectivamente as temperaturas registradas no 1º, no 2º e no 3º dias.

Saída

Carinha feliz, ou carinha triste.

Exemplos de Entrada	Exemplos de Saída
20 10 12	:)
10 20 18	:(
4 16 20	:(
4 10 20	:)
20 10 6	:)

20 16 4	:(
10 10 14	:)
10 10 2	:(

Objetivo: testar a assertividade do programa ClientesNoInverno.