

Conteúdo

1	Java, classes e polimorfismo	2
2	Arquivos	12

Lista 1: Java, classes e polimorfismo

1. Escreva um programa java que mostre ao usuário uma janela que contém um painel capaz de receber comandos com o mouse para definir o ponto central de um círculo e seu raio. Em quanto o usuário arrasta o mouse para desenhar o círculo, em uma barra na base da janela devem ser mostrados os valores do ponto central do círculo, o valor de seu raio, o valor de sua área e o valor de sua circunferência.
2. Escreva uma classe a ser acrescentada ao programa da questão 1, para gravar os valores correspondentes ao ponto central e ao raio de uma série de círculos definidos pelo usuário em uma janela que contém um painel capaz de receber comandos com o mouse para definir o ponto central de um círculo e seu raio. Em quanto o usuário arrasta o mouse para desenhar o círculo, em uma barra na base da janela devem ser mostrados os valores do ponto central do círculo, o valor de seu raio, o valor de sua área e o valor de sua circunferência.
3. Escreva um programa java que mostre ao usuário uma janela contendo um painel de desenho, uma barra de mensagens na parte inferior da janela e uma barra de botões na parte superior da janela. Nesta barra, devem ser apresentadas opções para selecionar os modos de desenho correspondentes a pontos, círculos e retângulos. Cada tipo de desenho deve ser armazenado em um vetor com pelo menos 500 posições. Na barra de mensagens, deve ser mostrado um texto informando as características de cada desenho conforme o usuário os realiza com o auxílio do mouse.
4. **Bits Trocados** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

As Ilhas Weblands formam um reino independente nos mares do Pacífico. Como é um reino recente, a sociedade é muito influenciada pela informática. A moeda oficial é o Bit; existem notas de B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00. Você foi contratado(a) para ajudar na programação dos caixas automáticos de um grande banco das Ilhas Weblands.

Tarefa

Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits.

Sua tarefa é escrever um programa que, dado o valor de Bits desejado pelo cliente, determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas entregues. Por exemplo, se o cliente deseja retirar B\$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B\$72,00, é necessário entregar uma nota de B\$50,00, duas de B\$10,00 e duas de B\$1,00.

Entrada

A entrada é um valor inteiro positivo V , que indica o valor solicitado pelo cliente. O final da entrada é indicado por $V = 0$.

Exemplo de Entrada

1
72
0

Saída

Para cada valor da entrada seu programa deve produzir uma linha na saída, na qual devem aparecer quatro inteiros I , J , K e L que representam o resultado encontrado pelo seu programa: I indica o número de cédulas de B\$50,00, J indica o número de cédulas de B\$10,00, K indica o número de cédulas de B\$5,00 e L indica o número de cédulas de B\$1,00.

Exemplo de Saída

0 0 0 1

1 2 0 2

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq V \leq 10000$ ($V = 0$ apenas para indicar o fim da entrada)

Observações

O seu programa deve possuir uma interface gráfica com o usuário, na qual um valor de saque é solicitado ao usuário e em seguida o número de notas de cada valor deve ser apresentado. Ao ser fornecido o valor 0 o programa deve ser encerrado.

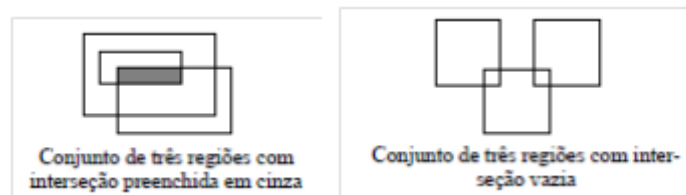
5. **Macaco-prego** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

O macaco-prego é um animal irrequieto e barulhento, merecedor também dos adjetivos desordeiro e despudorado. A sua cabeça, encimada por uma densa pelagem negra ou marrom-escura, semelhante a um gorro, torna seu aspecto inconfundível. Apesar de ser o macaco mais comum nas matas do país, uma de suas sub-espécies encontra-se seriamente ameaçada de extinção: o macaco-prego-do-peito-amarelo, que se distingue das demais pela coloração amarelada do peito e da parte anterior dos braços.

Um grande esforço foi feito pelos primatologistas para aumentar a população dos macacos-prego-do-peito-amarelo. Sabe-se que eles se alimentam de plantas, das quais consomem preferencialmente frutos e brotos. Alimentam-se também de muitos animais, preferencialmente lesmas, lagartas e rãs, e preferem as florestas mais densas. Para determinar o melhor local do país para criar uma nova reserva ambiental para os macacos-prego-do-peito-amarelo, o governo fez um levantamento das regiões no país onde as condições preferidas desses animais ocorrem: regiões de floresta densa, regiões com frutos, regiões com muitos brotos, etc. Ajude a salvar os macacos-prego-do-peito-amarelo.

Tarefa

As regiões propícias para o macaco-prego-do-peito-amarelo foram determinadas como retângulos cujos lados são todos verticais ou horizontais. Sua tarefa é encontrar o local ideal para a reserva ambiental, definida como a interseção de todas as regiões dadas.



As regiões foram divididas de tal forma que uma região não tangencia qualquer outra região. Assim, a interseção entre quaisquer duas regiões ou é um retângulo ou é vazia.

Entrada

Seu programa deve ler vários conjuntos de teste. A primeira linha de um conjunto de teste contém um inteiro não negativo, N , que indica o

número de regiões (o valor $N = 0$ indica o final da entrada). Seguem-se N linhas, cada uma contendo quatro números inteiros X , Y , U e V que descrevem uma região: o par X , Y representa a coordenada do canto superior esquerdo e o par U , V representa a coordenada do canto inferior direito de um retângulo.

Exemplo de Entrada

```
3
0 6 8 1
1 5 6 3
2 4 9 0
3
0 4 4 0
3 1 7 -3
6 4 10 0
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n ”, onde n é numerado a partir de 1. A segunda linha deve conter as coordenadas do retângulo de interseção encontrado pelo seu programa, no mesmo formato utilizado na entrada. Caso a interseção seja vazia, a segunda linha deve conter a expressão “nenhum”. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
2 4 6 3
Teste 2
nenhum
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)
 $-10000 \leq X \leq 10000$

$$-10000 \leq Y \leq 10000$$

$$-10000 \leq U \leq 10000$$

$$-10000 \leq V \leq 10000$$

6. **Quermesse** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

Os alunos do último ano resolveram organizar uma quermesse para arrecadar fundos para a festa de formatura. A festa prometia ser um sucesso, pois o pai de um dos formandos, Teófilo, dono de uma loja de informática, decidiu doar um computador para ser sorteado entre os que comparecessem. Os alunos prepararam barracas de quentão, pipoca, doces, ensaiaram a quadrilha e colocaram à venda ingressos numerados sequencialmente a partir de 1. O número do ingresso serviria para o sorteio do computador. Ficou acertado que Teófilo decidiria o método de sorteio; em princípio o sorteio seria, claro, computadorizado.

O local escolhido para a festa foi o ginásio da escola. A entrada dos participantes foi pela porta principal, que possui uma roleta, onde passa uma pessoa por vez. Na entrada, um funcionário inseriu, em uma lista no computador da escola, o número do ingresso, na ordem de chegada dos participantes. Depois da entrada de todos os participantes, Teófilo começou a trabalhar no computador para preparar o sorteio. Verificando a lista de presentes, notou uma característica notável: havia apenas um caso, em toda a lista, em que o participante que possuía o ingresso numerado com i , havia sido a i -ésima pessoa a entrar no ginásio. Teófilo ficou tão encantado com a coincidência que decidiu que o sorteio não seria necessário: esta pessoa seria o ganhador do computador.

Tarefa

Conhecendo a lista de participantes, por ordem de chegada, sua tarefa é determinar o número do ingresso premiado, sabendo que o ganhador é o único participante que tem o número do ingresso igual à sua posição de entrada na festa.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro positivo N que indica o número de participantes da festa. A linha seguinte contém a sequência, em ordem de entrada, dos N ingressos das pessoas que participaram da festa. O final da entrada é indicado quando $N = 0$. Para cada conjunto de teste da entrada haverá um único ganhador.

Exemplo de Entrada

```
4
4 5 3 1
10
9 8 7 6 1 4 3 2 12 10
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A segunda linha deve conter o número do ingresso do ganhador, conforme determinado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
3
Teste 2
10
```

(esta saída corresponde ao exemplo de entrada acima) **Restrições**

$0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)

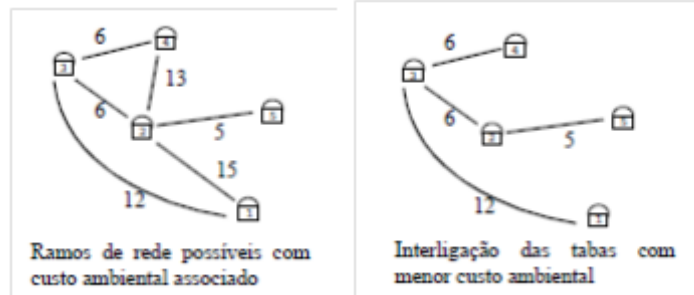
7. **Rede ótica** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

Os caciques da região de Tutuaçu pretendem integrar suas tribos à chamada "aldeia global". A primeira providência foi a distribuição de telefones celulares a todos os pajés. Agora, planejam montar uma rede de fibra ótica interligando todas as tabas. Esta empreitada requer que sejam abertas novas picadas na mata, passando por reservas de flora e fauna. Conscientes da necessidade de preservar o máximo possível o meio ambiente, os caciques encomendaram um estudo do impacto ambiental do projeto. Será que você consegue ajudá-los a projetar a rede de fibra ótica?

Tarefa

Vamos denominar uma ligação de fibra ótica entre duas tabas de um ramo de rede. Para possibilitar a comunicação entre todas as tabas

é necessário que todas elas estejam interligadas, direta (utilizando um ramo de rede) ou indiretamente (utilizando mais de um ramo). Os caciques conseguiram a informação do impacto ambiental que causará a construção dos ramos. Alguns ramos, no entanto, nem foram considerados no estudo ambiental, pois sua construção é impossível.



Sua tarefa é escrever um programa para determinar quais ramos devem ser construídos, de forma a possibilitar a comunicação entre todas as tabas, causando o menor impacto ambiental possível.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois números inteiros positivos N e M que indicam, respectivamente, o número de tabas e o número de ramos de redes possíveis. As tabas são numeradas de 1 a N . As M linhas seguintes contém três inteiros positivos X , Y e Z , que indicam que o ramo de rede que liga a taba X à taba Y tem impacto ambiental Z . Com os conjuntos de teste dados sempre é possível interligar todas as tabas. O final da entrada é indicado quando $N = 0$.

Exemplo de Entrada

```
3 3
1 2 10
2 3 10
3 1 10
5 6
1 2 15
1 3 12
2 4 13
2 5 5
3 2 6
```


3 4 6
0 0

Saída

Para cada conjunto de teste da entrada seu programa deve produzir uma lista dos ramos de redes que devem ser construídos. A lista deve ser precedida de uma linha que identifica o conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A lista é composta por uma sequência de ramos a serem construídos, um ramo por linha. Um ramo é descrito por um par de tabas X e Y, com $X \leq Y$. Os ramos de rede podem ser listados em qualquer ordem, mas não deve haver repetição. Se houver mais de uma solução possível, imprima apenas uma delas. O final de uma lista de ramos deve ser marcado com uma linha em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
1 2
1 3
Teste 2
1 3
2 3
2 5
3 4
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 100$ ($N = 0$ apenas para indicar o fim da entrada)

$1 \leq M \leq N(N - 1)/2$

$1 \leq X \leq 100$

$1 \leq Y \leq 100$

$1 \leq Z \leq 100$

8. **Saldo de gols** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

Hipólito é um torcedor fanático. Coleciona flâmulas, bandeiras, recortes de jornal, figurinhas de jogadores, camisetas e tudo o mais que se refira

a seu time preferido. Quando ganhou um computador de presente em uma festa, resolveu montar um banco de dados com os resultados de todos os jogos de seu time ocorridos desde a sua fundação, em 1911. Depois de inseridos os dados, Hipólito começou a ficar curioso sobre estatísticas de desempenho do time. Por exemplo, ele deseja saber qual foi o período em que o seu time acumulou o maior saldo de gols. Como Hipólito tem o computador há muito pouco tempo, não sabe programar muito bem, e precisa de sua ajuda.

Tarefa

É dada uma lista, numerada sequencialmente a partir de 1, com os resultados de todos os jogos do time (primeira partida: 3 x 0, segunda partida: 1 x 2, terceira partida: 0 x 5 ...). Sua tarefa é escrever um programa que determine em qual período o time conseguiu acumular o maior saldo de gols. Um período é definido pelos números de sequência de duas partidas, A e B, onde $A \leq B$. O saldo de gols acumulado entre A e B é dado pela soma dos gols marcados pelo time em todas as partidas realizadas entre A e B (incluindo as mesmas) menos a soma dos gols marcados pelos times adversários no período. Se houver mais de um período com o mesmo saldo de gols, escolha o maior período (ou seja, o período em que $B - A$ é maior). Se ainda assim houver mais de uma solução possível, escolha qualquer uma delas como resposta.

Entrada

Seu programa deve ler vários conjuntos de teste. A primeira linha de um conjunto de teste contém um inteiro não negativo, N, que indica o número de partidas realizadas pelo time (o valor $N = 0$ indica o final da entrada). Seguem-se N linhas, cada uma contendo um par de números inteiros não negativos X e Y que representam o resultado da partida: X são os gols a favor e Y os gols contra o time de Hipólito. As partidas são numeradas sequencialmente a partir de 1, na ordem em que aparecem na entrada.

Exemplo de Entrada

```
2
2 3
7 1
9
2 2
0 5
6 2
```

```
1 4
0 0
5 1
1 5
6 2
0 5
3
0 2
0 3
0 4
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter um par de inteiros I e J que indicam respectivamente a primeira e última partidas do melhor período, conforme determinado pelo seu programa, exceto quando o saldo de gols do melhor período for menor ou igual a zero; neste caso a segunda linha deve conter a expressão “nenhum”. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
2 2
Teste 2
3 8
Teste 3
nenhum
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)

$1 \leq A \leq N$

$A \leq B \leq N$

$0 \leq X \leq 50$

$0 \leq Y \leq 50$

Lista 2: Arquivos

1. Escreva um programa java que mostre ao usuário uma janela contendo um painel de desenho, uma barra de mensagens na parte inferior da janela e uma barra de botões na parte superior da janela. Nesta barra, devem ser apresentadas opções para selecionar os modos de desenho correspondentes a pontos, círculos e retângulos, um botão para gravar e outro para ler as informações de um arquivo utilizado para armazenar os dados do desenho. Cada tipo de desenho deve ser armazenado em um vetor com pelo menos 500 posições. Na barra de mensagens, deve ser mostrado um texto informando as características de cada desenho conforme o usuário os realiza com o auxílio do mouse. Ao clicar no botão para gravar, as informações de todos os desenhos feitos pelo usuário devem ser gravadas no arquivo, e ao clicar no botão para ler, as informações dos desenhos devem ser descartadas e substituídas pelas informações contidas no arquivo.
2. Escreva um programa java que solicite ao usuário o caminho para uma pasta em um dispositivo de armazenamento em massa e, a partir desta pasta, leia todos os nomes de arquivos e pastas aí contidas. Para cada pasta encontrada o processo deve ser repetido. Na interface do seu programa deve ser apresentado ao usuário o total de arquivos e de pastas encontradas e uma opção para que a estrutura encontrada seja mostrada de forma hierárquica ao usuário.
3. **Notas de alunos**
 - (a) Elabore um programa que armazene em um arquivo o nome e a nota de n alunos.
 - (b) Leia o arquivo com os nomes e as notas dos alunos e mostre:
 - o nome do aluno que obteve a maior nota;
 - o nome do aluno com a menor nota;
 - a média das notas da sala.
4. Faça um programa que seja capaz de ler os valores de uma matriz $n \times m$ do usuário e salve-a em um arquivo de texto. Depois, leia o arquivo e guarde os valores em uma nova matriz e mostre a matriz recuperada.
5. Faça um programa que seja capaz de ler os valores de uma matriz $n \times m$ do usuário e salve-a em um arquivo binário. Depois, leia o arquivo e guarde os valores em uma nova matriz e mostre a matriz recuperada.

6. Elabore um programa simples para criptografar um arquivo de texto. O programa deverá receber do usuário o nome do arquivo a ser criptografado e um valor inteiro de deslocamento. De forma que cada caractere será deslocado de acordo com o fator de deslocamento.

Exemplos:

Arquivo de entrada: "1 2 3 4"; Fator de deslocamento = 1; Saída: "2 3 4 5"

Arquivo de entrada: "1 2 3 4"; Fator de deslocamento = 3; Saída: "4 5 6 7"

Arquivo de entrada: "1 2 3 4"; Fator de deslocamento = 10; Saída: "11 12 13 14"

Salve o texto criptografado em um novo arquivo binário. Depois faça a operação inversa, leia o arquivo criptografado e utilize o fator de deslocamento para decifrar o texto.

7. Faça um verificador de respostas para auxiliar na correção de provas. Para isto, utilize um arquivo como gabarito, onde cada linha representa a resposta correta para uma questão; e outro arquivo, correspondente às respostas dadas por um aluno. Considere que existem 5 opções para cada questão da prova: A, B, C, D, E. Imprima a quantidade de acertos do aluno.

8. **Bits Trocados** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

As Ilhas Weblands formam um reino independente nos mares do Pacífico. Como é um reino recente, a sociedade é muito influenciada pela informática. A moeda oficial é o Bit; existem notas de B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00. Você foi contratado(a) para ajudar na programação dos caixas automáticos de um grande banco das Ilhas Weblands.

Tarefa

Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits.

Sua tarefa é escrever um programa que, dado o valor de Bits desejado pelo cliente, determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas

entregues. Por exemplo, se o cliente deseja retirar B\$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B\$72,00, é necessário entregar uma nota de B\$50,00, duas de B\$10,00 e duas de B\$1,00.

Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um número inteiro positivo V , que indica o valor solicitado pelo cliente. O final da entrada é indicado por $V = 0$. Exemplo de Entrada

```
1
72
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n ”, onde n é numerado a partir de 1. Na segunda linha devem aparecer quatro inteiros I , J , K e L que representam o resultado encontrado pelo seu programa: I indica o número de cédulas de B\$50,00, J indica o número de cédulas de B\$10,00, K indica o número de cédulas de B\$5,00 e L indica o número de cédulas de B\$1,00. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
0 0 0 1
Teste 2
1 2 0 2
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq V \leq 10000$ ($V = 0$ apenas para indicar o fim da entrada)

9. Meteoros (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

Em noites sem nuvens pode-se muitas vezes observar pontos brilhantes no céu que se deslocam com grande velocidade, e em poucos segundos

desaparecem de vista: são as chamadas estrelas cadentes, ou meteoros. Meteoros são na verdade partículas de poeira de pequenas dimensões que, ao penetrar na atmosfera terrestre, queimam-se rapidamente (normalmente a uma altura entre 60 e 120 quilômetros). Se os meteoros são suficientemente grandes, podem não queimar-se completamente na atmosfera e dessa forma atingem a superfície terrestre: nesse caso são chamados de meteoritos.

Zé Felício é um fazendeiro que adora astronomia e descobriu um portal na Internet que fornece uma lista das posições onde caíram meteoritos. Com base nessa lista, e conhecendo a localização de sua fazenda, Zé Felício deseja saber quantos meteoritos caíram dentro de sua propriedade. Ele precisa de sua ajuda para escrever um programa de computador que faça essa verificação automaticamente.

Tarefa

São dados:

- uma lista de pontos no plano cartesiano, onde cada ponto corresponde à posição onde caiu um meteorito;
- as coordenadas de um retângulo que delimita uma fazenda.

As linhas que delimitam a fazenda são paralelas aos eixos cartesianos. Sua tarefa é escrever um programa que determine quantos meteoritos caíram dentro da fazenda (incluindo meteoritos que caíram exatamente sobre as linhas que delimitam a fazenda).

Entrada

Seu programa deve ler vários conjuntos de testes. A primeira linha de um conjunto de testes quatro números inteiros X_1 , Y_1 , X_2 e Y_2 , onde (X_1, Y_1) é a coordenada do canto superior esquerdo e (X_2, Y_2) é a coordenada do canto inferior direito do retângulo que delimita a fazenda. A segunda linha contém um inteiro, N , que indica o número de meteoritos. Seguem-se N linhas, cada uma contendo dois números inteiros X e Y , correspondendo às coordenadas de cada meteorito. O final da entrada é indicado por $X_1 = Y_1 = X_2 = Y_2 = 0$.

Exemplo de Entrada

```
2 4 5 1
2
1 2
```

```
3 3
2 4 3 2
3
1 1
2 2
3 3
0 0 0 0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter o número de meteoritos que caíram dentro da fazenda. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
1
Teste 2
2
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$$0 \leq N \leq 10.000$$

$$0 \leq X \leq 10.000$$

$$0 \leq Y \leq 10.000$$

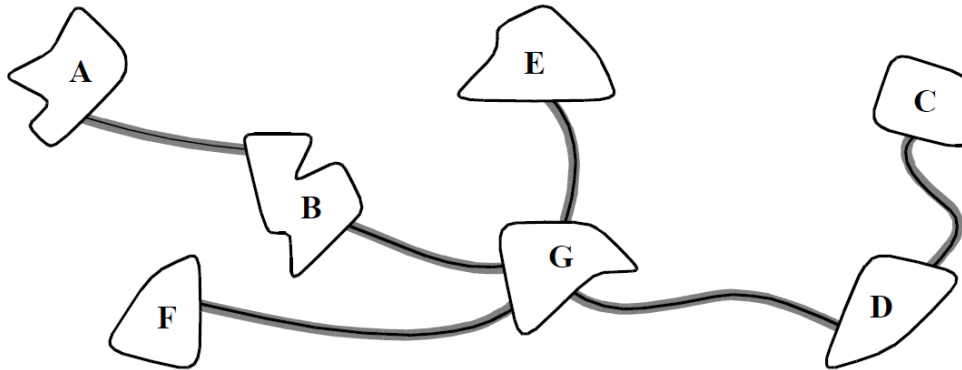
$$0 \leq X1 < X2 \leq 10.000$$

$$0 \leq Y2 < Y1 \leq 10.000$$

10. **Dengue** (Exercício retirado/adaptado da Olimpíada Brasileira de Informática.)

A Costa do Mosquito é um país pequeno mas paradisíaco. Todos os habitantes têm boas moradias, bons empregos, o clima é agradável, e os governantes são justos e incorruptíveis. O sistema de transporte público da Costa do Mosquito é composto de uma rede de linhas de trem. O sistema foi projetado de forma peculiar: existe um único percurso

Figura 1: Sete vilas interligadas por linhas de trem



ligando duas quaisquer vilas (esse percurso possivelmente passa por outras vilas). Por exemplo, na figura abaixo, que mostra um trecho do mapa da Costa do Mosquito, há apenas um percurso entre as vilas A e C, passando pelas vilas B, G e D. Uma tarifa fixa de M\$ 1,00 é cobrada por cada viagem entre vilas vizinhas; assim, para uma viagem de A a C o usuário gasta M\$ 4,00.

Devido a um inesperado surto de dengue, o Ministério da Saúde da Costa do Mosquito resolveu montar um Posto de Vacinação. Para evitar que habitantes gastem muito dinheiro para se deslocar até a vila onde ficará o Posto de Vacinação, o Ministério da Saúde decidiu que este deverá ser instalado em uma vila de forma que o gasto com transporte até o Posto, para os habitantes que gastarem mais, seja o menor possível (para o caso da figura abaixo a vila escolhida seria G).

Tarefa

Sua tarefa é escrever um programa que determine uma vila onde deve ser instalado o Posto de Vacinação. Esta vila deve ser tal que o custo com transporte, para os habitantes que tiverem maior custo, seja o menor possível. Note que devido à característica peculiar do sistema viário, ou haverá uma única vila que satisfaz essa restrição, ou haverá duas vilas que a satisfazem. No caso de existirem duas vilas apropriadas, qualquer uma delas serve como solução.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de testes contém um número inteiro N que indica a

quantidade de vilas do país. As vilas são numeradas de 1 a N. As N-1 linhas seguintes contêm, cada uma, dois inteiros positivos X e Y que indicam que a vila X tem um caminho que a liga diretamente com a vila Y, sem passar por outras vilas. O final da entrada é indicado por $N = 0$.

Exemplo de Entrada

```
2
1 2
7
1 2
2 5
7 4
7 2
4 6
3 4
1
0 0
```

Saída Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter o número da vila na qual deve ser instalado o Posto de Vacinação. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
1
Teste 2
7
Teste 3
1
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$$0 \leq N \leq 100$$

($N = 0$ apenas para indicar o final da entrada)

11. Escreva um programa em Java que permita escolher um arquivo, abri-lo em formato binário e apresentar em uma janela o seu conteúdo de forma tabular, com uma coluna de índice e colunas com os valores dos bytes do arquivo, em formato hexadecimal, seguindo as definições apresentadas na tabela a seguir:

Índice	Conteúdo do arquivo
0000	00 01 02 03 FF 00 FF 00 A2 3F 19 33
0012	00 01 02 03 FF 00 FF 00 A2 3F 19 33
0024	00 01 02 03 FF 00 FF 00 A2 3F 19 33
0036	00 01 02 03 FF 00 FF 00 A2 3F 19 33

12. Escreva um programa que permita ao usuário escrever uma sequência de bytes, podendo ele escolher entre digitar em formato decimal, hexadecimal ou binário, e em seguida os valores devem ser gravados em um arquivo, respeitando os valores digitados pelo usuário em formato digital. Deve ser possível também escolher gravar o arquivo em formato texto, de maneira que o valor de cada byte digitado seja separado por um espaço no arquivo, e seja sempre gravado em formato hexadecimal. Uma sugestão para a interface do programa é a utilização de dados tabulados, como apresentado na tabela a seguir:

Índice	Conteúdo do arquivo
	Binário
0000	00000000 01010101 10101010 11110000 00000000 01010101 10101010 11110000
0008	00000000 01010101 10101010 11110000 00000000 01010101 10101010 11110000
0016	00000000 01010101 10101010 11110000 00000000 01010101 10101010 11110000
0024	00000000 01010101 10101010 11110000 00000000 01010101 10101010 11110000

13. Escreva um programa que permita ao usuário escolher um arquivo, abri-lo em formato binário e apresentar em sua janela o conteúdo de forma tabular, sendo possível escolher entre a apresentação dos dados em formato binário, decimal, octal ou hexadecimal. Além de apresentar as informações, o programa deve permitir ao usuário alterar o conteúdo e em seguida salvar as alterações, no mesmo arquivo ou em outro a sua escolha. Também deve ser possível escolher gravar o conteúdo do arquivo em formato textual, conforme o apresentado em sua janela.

Na tabela a seguir é apresentada uma sugestão para a apresentação do conteúdo do arquivo na janela do programa:

Índice	Conteúdo do arquivo		
	Decimal		
0000	000 001 002 003	255 00 125 00	075 100 019 241
0012	000 001 002 003	255 00 123 00	023 100 019 250
0024	000 001 002 003	250 00 255 00	045 100 019 033
0036	000 001 002 003	200 00 255 00	125 200 019 033

14. Arquivo Bitmap

Escreva um programa em java que contenha uma classe capaz de ler um arquivo do tipo Bitmap (.BMP) do disco, uma segunda classe que seja capaz de apresentar o conteúdo do arquivo em um painel, para que o usuário possa visualizar a imagem contida no arquivo. **Descrição do formato de arquivo BMP:**

Todo arquivo BMP está dividido em 3 ou 4 partes, que são:

Cabeçalho de arquivo Contém a assinatura BM e informações sobre o tamanho e lay-out do arquivo BMP (disposição dos dados dentro do arquivo)

Cabeçalho de mapa de bits Contém as informações da imagem contida no arquivo. Define as dimensões, tipo de compressão (se houver) e informações sobre as cores da imagem.

Paleta ou mapa de cores (opcional) Somente estará presente em arquivos de imagens que usam 16 ou 256 cores (4 e 8 bits/pixel). Nas demais, em seu lugar, vem diretamente a parte seguinte: área de dados da imagem.

Área de dados da imagem contida no arquivo Dados que permitem a exibição da imagem propriamente dita, o dados dos pixels a serem exibidos. Podem ser com ou sem compressão.

Existem dentro da estrutura do BMP alguns campos ditos "Reservados", destinados a uso futuro, que sempre devem ser setados com ZERO. Outros dados são sempre idênticos ou fornecidos mais de uma vez no arquivo, para possível re-verificação e desvio de possíveis erros.

Cabeçalho de arquivo

Informações do arquivo (Tamanho: 14 bytes)

Campo	Tamanho em Bytes	Descrição
BfType	2	Assinatura do arquivo: os caracteres ASCII "BM" ou $(424D)_h$. É a identificação de ser realmente BMP.
BfSize	4	Tamanho do arquivo em Bytes.
BfReser1	2	Campo reservado 1; deve ser ZERO.
BfReser2	2	Campo reservado 2; deve ser ZERO.
BfOffsetBits	4	Especifica o deslocamento, em bytes, para o início da área de dados da imagem, a partir do início deste cabeçalho. Se a imagem usa paleta, este campo tem tamanho=14+40+(4 x NumeroDeCores). Se a imagem for true color, este campo tem tamanho=14+40=54.

Cabeçalho de mapa de bits

Informações da imagem (Tamanho: 40 bytes)

Campo	Tamanho em Bytes	Descrição
BiSize	4	Tamanho deste cabeçalho (40 bytes). Sempre $(28)_h$.
BiWidth	4	Largura da imagem em pixels.
BiHeight	4	Altura da imagem em pixels.
BiPlanes	2	Número de planos de imagem. Sempre 1.
BiBitCount	2	Quantidade de Bits por pixel (1,4,8,24,32). Este campo indica, indiretamente, ainda o número máximo de cores, que é 2 Bits por pixel.

Continua na próxima página.

Continuação da página anterior

Campo	Tamanho em Bytes	Descrição
BiCompress	4	Compressão usada. Pode ser: <ul style="list-style-type: none"> • 0 = BI_RGB → sem compressão • 1 = BI_RLE8 → compressão RLE 8 bits • 2 = BI_RLE4 → compressão RLE 4 bits
BiSizeImag	4	Tamanho da imagem (dados) em byte <ul style="list-style-type: none"> • Se arquivo sem compressão, este campo pode ser ZERO. • Se imagem em true color, será Tamanho do arquivo (Bfsize) menos deslocamento (BfOffSetBits)
BiXPPMeter	4	Resolução horizontal em pixels por metro.
BiYPPMeter	4	Resolução vertical em pixels por metro.
BiClrUsed	4	Número de cores usadas na imagem. Quando ZERO indica o uso do máximo de cores possível pela quantidade de Bits por pixel, que é 2 Bits por pixel.
BiClrImpor	4	Número de cores importantes (realmente usadas) na imagem. Por exemplo das 256 cores, apenas 200 são efetivamente usadas. Se todas são importantes pode ser ZERO. É útil quando for exibir uma imagem em 1 dispositivo que suporte menos cores que a imagem possui.

Paleta de cores

definição da tabela de cores (Tamanho: 4 bytes x Número de Cores)

Campo	Tamanho em Bytes	Descrição
Blue	1	Intensidade de Azul. De 0 a 255.
Green	1	Intensidade de Verde. De 0 a 255.
Red	1	Intensidade de Vermelho. De 0 a 255.
Reservado	1	Campo reservado deve ser ZERO sempre.

A tabela de cores só está presente em imagens com 256 cores ou menos. Para os demais tipos de BMP, vem em seu lugar a área de dados. No BMP a cor é representada de forma diferenciada dos demais formatos de arquivos de imagens.

A paleta é um vetor de bytes da estrutura RGBA, representando a intensidade de cada cor, através de 1 byte. Imagens de 8 bits por pixel, com no máximo 256 cores terão 256 posições na paleta; da mesma forma que imagens de 4 bits por pixel terão 16 posições e imagens de 1 bit por pixel terão 2 posições na paleta.

Área de dados da imagem:

Cor que cada pixel deve ser ligado ou esses dados comprimidos - Tamanho: campo BiSizeImg, do cabeçalho de informações da imagem.

Esta área do arquivo de imagens varia, conforme existência ou não de compressão. Para imagens sem compressão, os dados são armazenados em uma ordem sequencial, dentro do arquivo, que corresponde a posições na tela de vídeo. O primeiro pixel refere-se a posição inferior esquerda. O último pixel refere-se a posição superior direita. Como a orientação usual de um sistema de coordenadas cartesiano.

O valor lido nessa área de dados, se sem compressão, refere-se a cor do pixel de acordo com a tabela de cores (paleta).

Há, entretanto, a restrição de que cada linha deva ter N bytes, sendo N um número divisível por 4. Caso contrário, o BMP deve ser preenchido com bytes não válidos. Por exemplo, se a imagem tem 1 x 100 pixels em 8 bits/pixel, o BMP teria 1 byte válido em cada linha e mais 3 bytes que não tem qualquer significado.

No BMP monocromático, cada valor lido corresponde a uma entrada na paleta de cores. Se o bit for ZERO a cor é a da paleta[0]; caso contrário

a cor será a da paleta[1]. No BMP de 16 cores, cada 4 bits (meio byte) correspondem a uma entrada na paleta de cores. Por exemplo se o primeiro byte contiver (1F)h, o primeiro pixel tem a cor da paleta[0] e o segundo pixel a cor da paleta[15].

No BMP de 256 cores, cada byte (8 bits) correspondem a uma entrada na paleta de cores. Se o primeiro byte da área de dados contém (1F)h o primeiro pixel tem a cor da paleta[31].

No BMP true color (24 bits) cada sequência de 3 bytes correspondem a uma sequência Blue.Green.Red , isso é a composição da cor do pixel diretamente (não tendo neste caso paleta de cores). No true color de 32 bits, é a combinação de 4 bytes que estabelecerá a cor do pixel.