

**Maratona de Programação
2001
Caderno de Problemas**

Plantando abacaxis

Arquivo: abacaxi.c ou abacaxi.pas

Entrada: abacaxi.in

Um plantador de abacaxi do Hawaii está analisando o custo e a receita da produção de suas fazendas, tentando determinar qual foi o período de maior lucro de sua produção.

Para fazer a análise, o produtor anotou para cada uma de suas fazendas e para cada safra as despesas de sua produção e a receita que obteve com a venda do abacaxi produzido. O objetivo do produtor é determinar em que período teve o maior lucro acumulado, a fim de ter uma idéia melhor das estratégias de manejo que deve tomar no futuro.

A entrada

São dadas várias seqüências correspondentes a diversas fazendas do produtor. Cada seqüência é dada por uma linha contendo o número $0 < n < 1000$ de safras da seqüência, uma linha contendo n números inteiros não negativos correspondentes às receitas de cada safra e uma linha contendo n números inteiros não negativos correspondentes às despesas de cada safra. A entrada termina com um zero.

A saída

Para cada fazenda você deverá imprimir uma linha identificando a mesma (**Fazenda 1**, etc). A seguir, deverá vir uma mensagem dizendo que período consecutivo foi o de maior lucro naquela fazenda **Início i Fim f**, onde i e f são respectivamente o início e o fim do período de maior lucro da fazenda. Se houver vários períodos empatando, devolva o mais próximo do início da seqüência. Persistindo o empate, o menor período deve ser devolvido. Caso o produtor só tenha tido prejuízo, seu programa deverá imprimir a mensagem **0 produtor só teve prejuízo nesta fazenda**.

Exemplo

Entrada

```
5
23 37 82 46 32
13 40 75 62 19
8
14 22 29 41 72 25 68 90
22 19 21 45 70 46 66 80
3
2 4 5
8 8 8
0
```

Saída

```
Fazenda 1
Início 1 Fim 3
```

```
Fazenda 2
Início 7 Fim 8
```

```
Fazenda 3
0 produtor só teve prejuízo nesta fazenda
```

Driblando o depto de trânsito

Arquivo: `multas.c` ou `multas.pas`

Entrada: `multas.in`

O departamento de trânsito de Honolulu é conhecido por sua rigidez e inflexibilidade. Os habitantes da ilha, entretanto, não se esforçam muito em manter seus carros abaixo do limite de velocidade permitido, acarretando um número enorme de multas de trânsito e apreensões de carteiras de motorista. Quando um motorista atinge 21 pontos (ou mais) acumulados em sua carteira, a mesma é suspensa, e este é impedido de dirigir por um longo período.

A família Kawaalulu teve uma idéia para tentar minimizar o número de membros da família com a carteira suspensa. Como no Brasil o Departamento de Trânsito envia ao dono do carro uma notificação de multa, e solicita que o mesmo diga quem foi o motorista que cometeu a contravenção, para que os pontos correspondentes a ela sejam creditados em sua habilitação. A idéia dos Kawaalulu é informar ao Departamento de Trânsito o membro da família com menos pontos na carteira, para cada nova multa. Caso haja mais de um membro da família com o número mínimo de pontos, o com o nome lexicograficamente menor recebe a multa (você pode supor que todos têm nomes diferentes). Um programa que implementa esta idéia já tem vários interessados no estado e já vem preocupando as autoridades de trânsito havaianas que temem que o chamado “jeitinho havaiano” se espalhe por todas as ilhas do arquipélago.

Sua tarefa neste problema é implementar o sistema da família havaiana.

A entrada

São dadas várias famílias. Cada família é dada, na primeira linha, pelo número $0 < n < 20$ de membros da família que possuem habilitação. Nas n linhas seguintes são dados os nomes (cadeias com até 80 caracteres sem espaço em branco) de cada membro da família. Em seguida, são dados os pontos (inteiros positivos) das diversas multas recebidas. As multas de uma família terminam quando um número de pontos negativo é lido. A entrada termina com um zero.

A saída

Para cada família você deverá imprimir uma linha identificando a mesma (**Familia 1**, etc). A seguir, para cada multa, você deverá imprimir uma mensagem dizendo a quem deve ser atribuída aquela multa seguindo a regra da família. Caso a multa acarrete em suspensão da carteira, seu programa deverá informar isso.

Exemplo

Entrada

```
5
john
mary
bob
rick
donna
4
4
5
7
7
7
-2
3
paul
irene
```

charlie

5

5

5

5

5

4

7

7

4

4

4

4

-3

0

Saída

Familia 1

Multa 1 bob

Multa 2 donna

Multa 3 john

Multa 4 mary

Multa 5 rick

Multa 6 bob

Familia 2

Multa 1 charlie

Multa 2 irene

Multa 3 paul

Multa 4 charlie

Multa 5 irene

Multa 6 paul

Multa 7 paul

Multa 8 charlie

Multa 9 irene

Multa 10 irene

Multa 11 paul

Multa 12 charlie carteira suspensa

Amigo Secreto

Arquivo: `amigo.c` ou `amigo.pas`

Entrada: `amigo.in`

Uma brincadeira muito comum e tradicional nos finais de ano é a realização do *Amigo Secreto*. Familiares, amigos e funcionários são exemplos de grupos de pessoas que trocam presentes nesta época sob as regras do amigo secreto.

Caso você não conheça as regras, devemos primeiro perguntar-lhe: onde você esteve durante sua vida inteira? Sim, já que até mesmo os havaianos brincam de amigo secreto. Para sua grande felicidade, no entanto, as regras são muito simples e você pegará o espírito da coisa em poucos segundos.

O organizador da brincadeira começa confeccionando papeizinhos com os nomes dos participantes. Estes papeizinhos são dobrados de modo que ninguém consiga ler o nome dos participantes escritos sem que sejam abertos. Os nomes são então colocados em um saquinho e cada participante retira um, que será o amigo que ele deverá presentear.

Não é difícil perceber que o sorteio acima (ligeiramente antiquado), sem qualquer cuidado, permite que um participante acabe presenteando a si mesmo, ou que um grupo de participantes acabe isolado dos demais. Por exemplo, Steve pode retirar Danno que pode retirar Steve. Toda pessoa que já participou da brincadeira sabe que isto é muito chato.

Outra coisa comum em sorteios manuais é que ao retirar um nome de um inimigo declarado a pessoa o devolve imediatamente ao saquinho, ameaçando retirar-se da brincadeira. Um algoritmo de sorteio automático deve prever estas situações.

Sua tarefa neste problema é desenvolver um programa que realiza um sorteio entre os participantes de forma que todos pertençam a um único grupo (evitando assim os subgrupos isolados) e satisfaça a especificação de pares de participantes que não deverão presentear-se diretamente.

Observe que se houver muitas inimizades no grupo de “amigos”, poderá ser impossível resolver o problema.

A entrada

Seu programa deverá estar preparado para ser executado sobre diversas instâncias de teste.

Cada instância começa com um número inteiro positivo $n < 20$ que especifica o número de participantes. Nas n linhas seguintes são fornecidos os nomes de cada participante. O nome de um participante é composto somente por letras. Maiúsculas e minúsculas não devem ser diferenciadas e nomes compostos serão separados por um ou mais espaços.

A seguir, é fornecido um inteiro $m \geq 0$ que é o número de pares de participantes que não poderão presentear-se diretamente. Nas m linhas seguintes são especificados os pares no formato `nome - nome`. Você pode supor que ambos são nomes válidos (especificados anteriormente) e pode haver qualquer número de espaços entre os nomes e o separador.

Todas as instâncias estão no mesmo arquivo de entrada. A entrada termina com um zero.

A saída

Para cada instância do arquivo de entrada seu programa deverá imprimir uma linha identificando a mesma (`Amigo Secreto 1`, etc). No caso de existir uma atribuição de presentes que satisfaça as restrições seu programa deverá imprimir em ordem alfabética para cada um dos participantes quem ele deve presentear (`nome -> nome`). Caso não haja uma solução seu programa deverá imprimir a mensagem `Nao e possivel resolver o problema`.

Exemplo

Entrada

```
5
joao da silva
manoel antonio
```

joaquim jose
maria joaquina
antonio maria
3
joao da silva - antonio maria
joaquim jose - antonio maria
joao da silva - maria joaquina
3
aloha
danno
steve
2
steve - danno
aloha - danno
0

Saída

Amigo Secreto 1
antonio maria -> manael antonio
joao da silva -> joaquim jose
joaquim jose -> maria joaquina
manael antonio -> joao da silva
maria joaquina -> antonio maria

Amigo Secreto 2
Nao e possivel resolver o problema

Calor no Hawaii

Arquivo: max.c ou max.pas

Entrada: max.in

A onda de calor no Hawaii está muito grande, levando as autoridades locais a se preocupar enormemente com as altas temperaturas. Só no último mês foram cancelados 3 luaus noturnos, uma vez que todos os participantes estavam dentro do mar e se recusavam a ir para a praia.

O governador do Hawaii resolveu então fazer um sistema de controle de picos de calor. Um pico de calor ocorre quando a temperatura média do dia foi estritamente maior que a do dia anterior e do dia seguinte. Para o início e o fim do período, basta que a temperatura seja maior que o do dia vizinho existente. Sua tarefa neste problema é contar o número de picos ocorridos num determinado período.

A entrada

São dadas várias seqüências de temperatura. Cada seqüência é dada por uma linha contendo o número $n > 0$ de termos da seqüência, e, na linha seguinte, os n números reais correspondentes às médias de temperatura. A entrada termina com um zero.

A saída

Para cada seqüência você deverá imprimir uma linha identificando a mesma (**Sequencia 1**, etc). A seguir deverá vir uma mensagem **No período ocorreram k picos de calor**.

Exemplo

Entrada

```
12
1.5 3.4 15.2 7.5 19.0 11.3 13.9 15.1 17.3 19.4 21.9 23.8
3
10.5 7.2 13.0
4
2.0 2.0 2.0 2.0
0
```

Saída

```
Sequencia 1
No periodo ocorreram 3 picos de calor
```

```
Sequencia 2
No periodo ocorreram 2 picos de calor
```

```
Sequencia 3
No periodo ocorreram 0 picos de calor
```

Ganhando no câmbio

Arquivo: cambio.c ou cambio.pas

Entrada: cambio.in

Até serem anexadas aos Estados Unidos as ilhas havaianas eram bastante independentes. Cada ilha tinha sua própria moeda, língua e religião.

O comércio entre as ilhas levou à necessidade de se estabelecer taxas de câmbio entre as diversas moedas. Devido às dificuldades de comunicação da época, muitas vezes estas taxas acabavam defasadas, acarretando que um especulador pudesse ganhar dinheiro em uma seqüência de trocas. Por exemplo, se 1 quicklu (moeda da ilha principal) compra 0.75 klitlu (moeda de Oahu), 1 klitlu compra 2 daltlulu (moeda de Molokai) e 1 daltlulu compra 0.7 quicklu, um especulador esperto poderia a partir de 1 quicklu acabar com 1.05 quicklu fazendo as três operações de câmbio em seguida.

Sua tarefa é determinar uma subsequência de trocas que maximize o lucro do especulador. Ou seja, para cada conjunto de moedas $\{c_1, c_2, \dots, c_n\}$, seu programa deverá indicar uma seqüência de k operações de câmbio a partir da primeira moeda que maximiza o lucro obtido, ou seja, se a tabela de taxas de câmbio for R (assim, na posição (i, j) da tabela R temos quanto da moeda j compramos com uma unidade da moeda i), desejamos maximizar o valor de

$$R[c_1, c_{i_1}] * R[c_{i_1}, c_{i_2}] * \dots * R[c_{i_{k-1}}, c_{i_k}] * R[c_{i_k}, c_1].$$

A entrada

São dadas várias entradas. Cada entrada inicia com o número $0 < n < 100$ de moedas. Em seguida vêm n linhas. A linha i tem n números reais, correspondente à taxa de câmbio da moeda i para cada uma das outras moedas do arquipélago. A entrada termina com um zero.

A saída

Para cada conjunto de moedas seu programa deverá identificá-lo (**Instancia 1**, etc). Em seguida, seu programa deverá indicar uma seqüência de operações de câmbio (sem repetir moedas) começando e terminando na primeira moeda que maximiza o lucro obtido: 1 -> i -> ... -> j -> 1 (veja exemplo abaixo).

Exemplo

Entrada

```
3
1 0.75 1.43
1.33 1 2
0.7 0.5 1
4
1 2 3 4
0.75 1 0.33 1.25
0.33 3.5 1 0.8
0.25 0.7 0.15 1
0
```

Saída

Instancia 1

1 -> 2 -> 3 -> 1

Instancia 2

1 -> 3 -> 2 -> 1

Turistas e Tubarões

Arquivo: `tt.c` ou `tt.pas`

Entrada: `tt.in`

Como você já deve ter ouvido falar, as praias havaianas, além de lindas, são ótimas para a prática de surfe, body boarding, knee boarding e também para o famoso *jacaré* (ou body surfe). Diversos turistas viajam para o Havaii todos os anos em busca da onda perfeita. Infelizmente, nos últimos meses, esta busca tornou-se um pouco perigosa.

Devido a modificações realizadas no ecossistema do oceano pacífico (basicamente pelo set de filmagens de Pearl Harbor), muitos tubarões emigraram de seus nichos e encontraram nas águas havaianas um lugar tranqüilo e de fácil obtenção de comida.

Preocupadas com os freqüentes ataques aos turistas, as autoridades havaianas resolveram instalar um sistema de monitoramento via satélite que visa detectar a presença de tubarões em áreas próximas a grupos de banhistas. Um dos módulos deste sistema consiste em verificar se um tubarão encontra-se entre um grupo de turistas. Este módulo é de vital importância, pois as autoridades pretendem enviar ambulâncias para o local caso a presença do terror dos mares seja verificada. Infelizmente, as chances de que nenhuma pessoa seja atacada é muito baixa.

Sua tarefa neste problema é ajudar as autoridades havaianas implementando o referido módulo. Neste caso, entendemos que um tubarão está em posição ameaçadora para o grupo de turistas se ele se encontrar dentro (estritamente) da região delimitada pelos banhistas.

A entrada

Cada instância do problema possui a estrutura que segue. Na primeira linha é fornecido um número inteiro não negativo $n < 100$ que representa o número de turistas na água. Nas n linhas seguintes são fornecidos pares de inteiros que representam localizações geográficas dos turistas (abscissa, ordenada). Na linha seguinte é fornecido um inteiro não negativo $m < 500$ que representa o número de tubarões detectados pelo satélite. Nas m linhas seguintes são fornecidos pares de inteiros que representam localizações geográficas dos tubarões (abscissa, ordenada). Todas as instâncias encontram-se no mesmo arquivo de entrada. Um valor igual a zero para n indica o final dos testes.

A saída

Para cada instância do arquivo de entrada, seu programa deve exibir uma mensagem identificando-a (*Situacao 1*, etc). Na linha seguinte, dependendo da situação, uma das seguintes mensagens deverá ser impressa:

- Alerta, presença de tubarão entre grupo de turistas.
- Grupo seguro.

Exemplo

Entrada

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

4 5
1
-1 0
0

Saída

Situação 1:

Alerta, presença de tubarão entre grupo de turistas.

Situação 2:

Grupo seguro.

Hawaianas, as legítimas

Arquivo: `ilhas.c` ou `ilhas.pas`

Entrada: `ilhas.in`

Todo turista que chega ao Hawaii se encanta com a beleza de suas ilhas, e sonha em poder conhecê-las durante os dias de sua permanência. O meio de transporte principal entre as ilhas é o serviço de caiaques-táxis. Estas taxistas (sim, só mulheres podem trabalhar nos caiaques) levam turistas entre as ilhas, mas seguem rotas estabelecidas há séculos e se recusam por motivos religiosos a fazer outros caminhos que não sejam os estabelecidos por seus antepassados. E note que os caminhos são orientados. Pode ser que ir da ilha i para a ilha j seja uma rota permitida e ir de j para i não seja. Os deuses hawaianos são realmente estranhos...

Outra particularidade das caiaque-taxistas hawaianas é que elas só aceitam levar o turista de uma ilha a outra se ela tiver certeza de que poderá retornar ao ponto de partida utilizando as rotas determinadas pelos deuses hawaianos.

Um turista alemão em visita ao Hawaii desconfiou de uma taxista hawaiana, achando que a mesma não era legítima. Resolveu então perguntar a ela se fazia alguns itinerários, para resolver sua dúvida. Num primeiro passo, o alemão repartiu as ilhas em grupos maximais de tal forma que todas as ilhas dentro de um grupo são atingíveis mutuamente. Com essa partição ele pôde formular suas perguntas e testar se sua hawaiana era legítima.

Sua tarefa é auxiliar o turista alemão em sua verificação, repartindo o arquipélago em grupos de ilhas que satisfazem as regras dos deuses aplicadas às caiaque-taxistas.

A entrada

São dadas várias instâncias. Cada instância é dada, na primeira linha, pelo número $0 < n < 20$ de ilhas no arquipélago (que são numeradas de 1 a n) e $0 < m < 400$ rotas possíveis entre as ilhas. Nas m linhas a seguir são dadas as possíveis rotas, através da ilha de partida e chegada de cada rota. A entrada termina com uma linha contendo dois zeros.

A saída

Para cada instância você deverá imprimir uma linha identificando a mesma (**Arquipélago 1**, etc). A seguir, você deverá imprimir os grupos de ilhas que podem ser início e fim de possíveis passeios conforme requisitado pelo turista alemão, identificados por **Grupo 1**, **Grupo 2**, etc.

Exemplo

Entrada

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

5 6
6 7
7 4
7 8
0 0

Saída

Arquipelago 1
Grupo 1: 1 2 3 4
Grupo 2: 5 6 7

Arquipelago 2
Grupo 1: 1 2 3
Grupo 2: 4 5 6 7
Grupo 3: 8