

ToPAS'22

Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores

<http://topas.dcc.fc.up.pt>

Conjunto de Problemas



Faculdade de Ciências da Universidade do Porto

6 de maio de 2022

Este conjunto de problemas deverá conter oito (8) problemas e dezoito (18) páginas.
Se faltar algum problema, por favor avise a organização.

Edição realizada em colaboração com:



Departamento de Engenharia Eletrónica e Informática, Faculdade
de Ciências e Tecnologia, Universidade do Algarve



Departamento de Informática, NOVA School of Science and
Technology | FCT NOVA, Universidade NOVA de Lisboa

ToPAS'22

Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores – FCUP
6 de maio de 2022

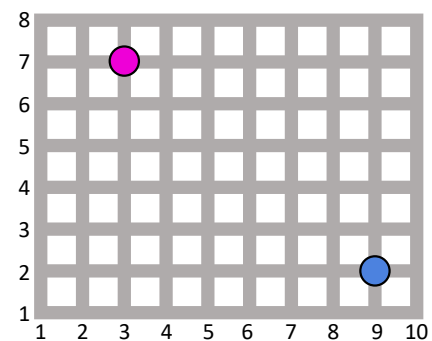
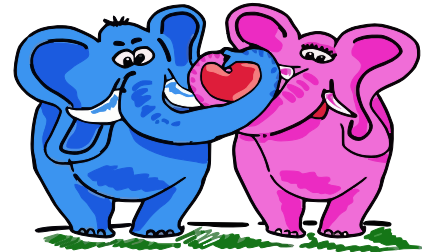
Conteúdo

Problema A: Elephantland	3
Problema B: Quem almoça de graça?	5
Problema C: Convites para os amigos	7
Problema D: Trilhos de números	9
Problema E: Shallow fake sunflowers	11
Problema F: Medições adocicadas	13
Problema G: Validar códigos de livros	15
Problema H: Atenção à tensão	17

Elephantland

A Walt Disney Pictures já revelou o tema do seu novo filme de animação, *Elephantland*. A história começa com um gigantesco incêndio numa savana africana. Para salvar os animais, os humanos transportam-nos para Manhattan, abandonando esta zona de Nova Iorque. Os protagonistas, Hanno e Kandula, são dois jovens elefantes apaixonados, que têm um papel crucial na sobrevivência de várias espécies no novo e hostil habitat.

Quando chegam a Manhattan, o primeiro desafio que enfrentam é reencontrarem-se. Mas têm de contornar os blocos de prédios, que parecem dispostos numa grelha regular, como se ilustra na figura à direita. Se o Hanno estiver no cruzamento da segunda avenida com a nona rua e a Kandula estiver no cruzamento de sétima avenida com a terceira rua, têm de percorrer, no total, pelo menos 11 blocos de prédios. Podem fazê-lo de muitas maneiras. Por exemplo, o Hanno pode-se deslocar 6 blocos na segunda avenida, até à terceira rua, e a Kandula pode andar 5 blocos na terceira rua, até à segunda avenida, encontrando-se no cruzamento da segunda avenida com a terceira rua.



Tarefa

Escreva um programa que, dados os cruzamentos de Manhattan onde estão o Hanno e a Kandula, calcula o número mínimo de blocos de prédios que os elefantes têm de andar, no total, para se encontrarem.

Input

O input tem duas linhas, cada uma com dois inteiros, A e R , que representam o cruzamento da avenida número A com a rua número R . A primeira linha refere-se ao Hanno e a segunda à Kandula.

Restrições

$1 \leq A \leq 100$ Número de ordem de uma avenida

$1 \leq R \leq 500$ Número de ordem de uma rua

Output

O output tem uma única linha, com um inteiro que representa o número mínimo de blocos de prédios que os elefantes têm de andar, no total, para se encontrarem.

Exemplo 1

Input

2 9

7 3

Output

11

Exemplo 2

Input

5 2

5 42

Output

40

Quem almoça de graça?

Um grupo de amigos gosta tanto de jogos que, quando almoçam juntos, usam o seguinte jogo para decidir quem é que nesse dia não paga o almoço. Cada um dos amigos escolhe um número, tendo uma forma de garantir que todos os números escolhidos são diferentes. Depois, fixam à sorte um número. Ganha quem tiver escolhido o número maior abaixo ou igual ao número fixado. Se todos tiverem escolhido números acima do número fixado, nesse dia ninguém tem direito a almoço grátis.



Tarefa

Escreva um programa que, dados o número fixado e o número escolhido por cada um dos amigos, escreva na consola: “No **free lunch**”, se ninguém ganhou; o número vencedor, caso alguém tenha ganho o jogo (e o almoço).

Input

A primeira linha do input tem um inteiro, F , que representa o número fixado por todos. A segunda linha tem um inteiro, N , que denota o número de amigos que almoçam juntos. Seguem-se N linhas, cada uma com um inteiro, e , que é o número escolhido por um dos amigos. Garante-se que todos os números escolhidos pelos amigos são diferentes.

Restrições

$1 \leq F \leq 99$ Número fixado

$1 \leq N \leq 10$ Número de amigos

$1 \leq e \leq 99$ Número escolhido por um amigo

Output

O output tem uma única linha com: o número vencedor, se algum amigo ganhou o almoço; “No **free lunch**” (sem aspas), se ninguém ganhou.

Exemplo 1

Input

12
3
3
7
8

Output

8

Exemplo 2

Input

14

4

31

17

18

22

Output

No free lunch

Exemplo 3

Input

20

5

31

17

18

22

21

Output

18

Convites para os amigos

O João faz parte de uma banda e em cada concerto tem direito a um conjunto de bilhetes para oferecer aos amigos. Antes dos concertos, vários amigos do João pedem-lhe bilhetes, não só para eles mas também para irem com amigos. O João tenta satisfazer os pedidos dos amigos, dando-lhes os bilhetes que pediram. Mas, como o número de convites que o João tem é limitado, pode haver amigos que recebem menos bilhetes do que



os que pediram. Como tudo se sabe, para que ninguém fique chateado, o João quer ser *justo*: não pode haver nenhum amigo que não tenha os bilhetes que pediu e que saiba que há outro amigo que tem mais bilhetes do que ele. Por exemplo, se o José pediu 10 bilhetes, mas o João só lhe consegue dar 5, não pode haver outro amigo que tenha mais do que 5 bilhetes. Outro exemplo é o caso extremo de não haver bilhetes suficientes sequer para os amigos do João. Nesse caso, o João prefere não dar bilhetes a ninguém.

O João gostava de ter um programa que o ajudasse a distribuir os convites pelos amigos. O objetivo do João é dar o maior número possível de bilhetes, sendo justo.

Tarefa

Escreva um programa que, dados o número de bilhetes que o João tem para oferecer e o número de bilhetes que cada um dos seus amigos pediu, calcule o número de bilhetes que o João deve dar a cada amigo.

Input

A primeira linha do input tem um inteiro, B , que representa o número de bilhetes que o João tem para oferecer. A segunda linha tem um inteiro, N , que denota o número de amigos do João. Seguem-se N linhas, cada uma com um inteiro, p , que é o número de bilhetes que um dos amigos do João lhe pediu.

Restrições

- $1 \leq B \leq 400$ Número de bilhetes que o João tem
- $1 \leq N \leq 30$ Número de amigos
- $1 \leq p \leq 25$ Número de bilhetes pedido por um amigo

Output

O output tem uma linha por cada amigo, pela ordem definida no input. Cada linha tem o número de bilhetes que o João deve dar ao respetivo amigo.

Exemplo 1**Input**

12
3
3
7
8

Output

3
4
4

Exemplo 2**Input**

15
4
3
4
2
1

Output

3
4
2
1

Trilhos de números

A Mariazinha gosta muito de labirintos e está a aprender os números. A professora de matemática deu-lhe vários cartões de uma atividade em que ela tem de descobrir um trilho de um algarismo num labirinto. A figura da direita tem dois desses cartões: no cartão de cima, pretende-se encontrar o trilho do número 1 para a abelha; no cartão de baixo, quer-se descobrir o trilho do número 2 para a borboleta. O trilho é um caminho composto por ocorrências vizinhas do número assinalado, que começa ao lado do inseto e termina junto à flor.

EDUCAÇÃO E TRANSFORMAÇÃO

	2	3	7	0	2	1
	1	1	4	8	1	2
2	3	2	1	7	5	4
1	1	1	1	2	6	7
1	9	2	4	7	2	
1	1	1	1	1	1	

EDUCAÇÃO E TRANSFORMAÇÃO

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

Para simplificar o problema, os labirintos são retangulares e o trilho começa sempre na posição do canto superior esquerdo. A Mariazinha tem de desenhar uma bola à volta desse número. Para avançar, a Mariazinha tem de colocar uma bola noutra ocorrência do mesmo número que ainda não esteja dentro de uma bola e que se encontre imediatamente acima (C), à direita (D), abaixo (B) ou à esquerda (E) da última bola desenhada. O trilho só está completo quando já não for possível avançar. A descrição do trilho corresponde à sequência de letras ‘C’, ‘D’, ‘B’ e ‘E’ que identificam os movimentos efetuados pelo inseto.

A figura abaixo tem um exemplo (que corresponde ao Exemplo 3). Em (a) apresenta-se o labirinto, que tem 3 linhas e 5 columnas. Em (b) mostra-se a primeira bola desenhada pela Mariazinha (à volta do número 2). A construção do trilho continua de (c) a (h): em cada um destes passos há uma nova bola e uma nova letra na descrição do caminho. A descrição do trilho é BBDDCD e a soma dos números do trilho (os que estão dentro de bolas) é 14.

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(a)

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(b)

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(c) B

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(d) BB

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(e) BBD

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(f) BBDD

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(g) BBDDC

2	3	4	5	6
2	4	2	2	7
2	2	2	3	2

(h) BBDDCD

Tarefa

Faça um programa que, dado um labirinto, escreva na consola a descrição do trilho e a soma de todos os números do trilho.

Garante-se que a Mariazinha nunca tem várias alternativas: em cada passo, ou pode desenhar uma bola numa única posição ou já não pode desenhar mais bolas. Também se garante que o trilho tem, pelo menos, dois números (logo, a sua descrição tem alguma letra).

Input

Na primeira linha existem dois números inteiros, L e C , que representam, respetivamente, o número de linhas e o número de colunas do labirinto. As L linhas seguintes descrevem o labirinto. Cada uma dessas linhas tem C algarismos de 1 a 9, separados por um espaço.

Restrições

- $1 \leq L \leq 50$ Número de linhas do labirinto
- $1 \leq C \leq 50$ Número de colunas do labirinto
- $2 \leq L \times C$ Número de algarismos no labirinto

Output

A primeira linha do output tem a descrição do trilho (que é uma sequência não vazia de letras maiúsculas 'C', 'D', 'B' e 'E'). A segunda linha tem a soma de todos os números do trilho.

Exemplo 1

Input

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

Output

```
DDDBBEEEBBDDDDDC
16
```

Exemplo 2

Input

```
4 5
9 9 4 9 9
2 9 2 9 7
2 9 2 9 1
2 9 9 9 1
```

Output

```
DBBBDDCCCD
99
```

Exemplo 3

Input

```
3 5
2 3 4 5 6
2 4 2 2 7
2 2 2 3 2
```

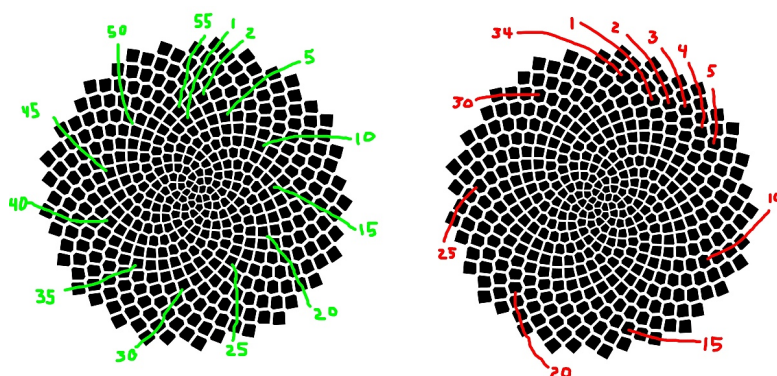
Output

```
BBDDCD
14
```

Shallow fake sunflowers

Nas redes sociais humanas proliferam as *deep fakes*. Nas dos girassóis o problema ainda são as falsificações grosseiras (*shallow fakes*) de fotografias de girassóis famosos.¹ Felizmente os *hackers* parecem ignorar algo importante sobre a disposição das sementes de girassol no centro das flores.

As sementes estão alinhadas em espirais, umas que evoluem a partir do centro no sentido horário (a verde na figura em baixo, à esquerda) e outras no sentido contrário (a vermelho na figura em baixo, à direita). E há uma relação entre o número de espirais nos dois sentidos: são sempre dois números consecutivos na sequência de Fibonacci! No exemplo ilustrado nas figuras, há 55 espirais no sentido horário e 34 espirais no sentido anti-horário.



A sequência de Fibonacci começa pelos números 0 e 1; ou seja, $F_0 = 0$ e $F_1 = 1$. Os valores seguintes da sequência são dados pela fórmula recursiva $F_n = F_{n-1} + F_{n-2}$ (i.e., para $n \geq 2$, o valor de F_n corresponde à soma dos dois valores anteriores). Os primeiros valores da sequência são os seguintes:

0 1 1 2 3 5 8 13 21 34 55 ...

Tarefa

A GiraMeta — a empresa que controla as principais redes sociais de girassóis — contratou-nos para desenvolver um programa que deteta *shallow fakes* nas fotografias publicadas. Já temos o módulo que conta o número de espirais em cada um dos sentidos — horário e anti-horário. Alguns girassóis são transgênicos e têm imensas espirais! Precisamos agora de um programa que determine quais são os pares que correspondem a girassóis verdadeiros. Ou seja, cujos números de espirais em cada sentido são números consecutivos da sequência de Fibonacci.

¹A imagem da direita é de um quadro de van Gogh (c.f. <https://creativecommons.org/licenses/by/3.0>, https://commons.wikimedia.org/wiki/File:Van_gogh,_girasoli,_188_o_1899,_02.JPG).

Input

A primeira linha do input contém um inteiro, N , que é o número de fotografias a analisar. Seguem-se N linhas, cada uma com dois inteiros, A e B , que indicam os números de espirais em sentido horário e anti-horário da fotografia de um girassol.

Restrições

$1 \leq N \leq 200$ Número de fotografias

$0 \leq A < 20\,000$ Número de espirais no sentido horário de uma fotografia

$0 \leq B < 20\,000$ Número de espirais no sentido anti-horário de uma fotografia

Output

O output tem uma linha por cada fotografia, pela ordem definida no input. Cada linha tem uma palavra, que classifica os valores correspondentes do input. Essa palavra é: “OK” (sem aspas), se os dois valores forem números consecutivos da sequência de Fibonacci; “FAKE” (sem aspas), no caso contrário. De notar que os valores **não** têm que estar por ordem crescente para serem números consecutivos da sequência de Fibonacci.

Exemplo 1

Input

3
1 1
3 5
2 5

Output

OK
OK
FAKE

Exemplo 2

Input

7
13 21
13 22
21 13
34 13
34 21
21 21
21 34

Output

OK
FAKE
OK
FAKE
OK
FAKE
OK

Medições adocicadas

Ufa, mas que calor! No refeitório da escola, a Mariana está a fazer experiências com sumo de laranja. Em cada experiência, ela coloca um copo graduado sobre a prateleira da máquina do sumo e abre a torneira. A cada segundo, o nível do sumo no copo sobe uma quantidade **inteira positiva** K . Repare que o copo pode não começar vazio. Assim, sendo o nível que o sumo apresentava inicialmente no copo um **inteiro não negativo** I , no momento em que a Mariana abre a torneira o nível do sumo é I , um segundo depois é $I + K$, dois segundos depois é $I + 2K$, e assim por diante.

Após um intervalo de tempo pré-determinado, a experiência acaba. Pode considerar que a capacidade do copo é suficiente para conter todo o sumo durante a experiência, isto é, em nenhum momento o conteúdo do copo transborda.

O objetivo da Mariana é, a cada segundo, observar o nível de sumo que o copo marca e anotá-lo no seu caderno. Acontece que, no final de cada experiência, um pouco de sumo caiu sobre o caderno, tornando alguns dos valores da sequência ilegíveis. Que trapalhada! Se ao menos a Mariana tivesse rabiscado em algum lado os valores de I e de K ...

É capaz de ajudar a Mariana a confirmar que a sequência por ela apontada antes de sujar o caderno com sumo poderia realmente ser válida ou dizer se, com o calor que fazia, ela se enganou?



Tarefa

Escreva um programa que, dada a sequência de valores apontados pela Mariana no seu caderno, possivelmente com alguns valores ilegíveis, determina se a sequência original poderia ser válida.

Input

A primeira linha do input contém um inteiro, E , que corresponde ao número de experiências (independentes) realizadas. Para cada experiência, é fornecida a seguinte informação:

- A primeira linha relativa a uma experiência contém um inteiro, N , que é o tamanho da sequência apontada no caderno.
- A segunda linha contém uma sequência de N inteiros, separados por um espaço. Cada elemento da sequência é um número entre -1 e $1\,000\,000$. Os elementos que ficaram ilegíveis com o sumo são representados com o valor -1 .

Restrições

$1 \leq E \leq 10$ Número de experiências
 $1 \leq N \leq 10\,000$ Tamanho de uma sequência

Output

O output tem uma linha por cada experiência, pela ordem definida no input. Cada linha tem a palavra: “**Possivel**” (sem aspas nem acento), se houver pelo menos uma forma de a respetiva sequência original ser válida; “**Impossivel**” (sem aspas nem acento), no caso contrário.

Exemplo

Input

```
5
3
4 -1 6
3
4 5 6
3
4 -1 4
5
-1 -1 4 6 -1
3
-1 -1 -1
```

Output

```
Possivel
Possivel
Impossivel
Possivel
Possivel
```

Explicação

- Na primeira experiência, o sumo tornou ilegível o segundo elemento da sequência, que poderia ser 5.
- Na segunda experiência, o sumo não tornou ilegível nenhum dos elementos da sequência e ela é válida, com $I = 4$ e $K = 1$.
- Na terceira experiência, se os valores apontados estivessem corretos, a torneira teria de ter estado fechada durante a experiência.
- Na quarta experiência, a sequência original poderia ser 0 2 4 6 8.
- Na quinta experiência, 7 20 33 é uma das sequências originais válidas possíveis.

Validar códigos de livros

O João voluntariou-se para ajudar na biblioteca da escola, que está a fazer um levantamento de todos os seus livros. A tarefa do João consiste em registar os livros existentes usando o código ISBN13 de cada livro.

O código ISBN13 é um código internacional que permite a identificação de livros, existindo um único código para cada edição e variação de um livro. É um código numérico constituído por 13 dígitos, em que o dígito mais à direita é o dígito de controlo/verificação de erro, sendo o seu valor calculado a partir dos outros 12 dígitos.

O valor do dígito de controlo é calculado seguindo os seguintes passos:

1. Cada um dos primeiros 12 dígitos (1 a 12) é multiplicado por 1 e por 3 alternadamente, ou seja, os dígitos das posições ímpares são multiplicados por 1 e os das posições pares por 3; esses produtos são todos somados;
2. Calcula-se o resto da divisão inteira da soma (obtida no passo 1) por 10;
3. Subtrai-se a 10 o valor do resto da divisão (obtido no passo 2); o resultado desta subtração é um número inteiro entre 1 e 10;
4. O dígito de controlo é 0, se o valor obtido no passo 3 for 10; no caso contrário, é o resultado da subtração.



Tarefa

Para garantir que não se enganou a registar o código ISBN13, o João quer ter um programa para verificar se o código registado é um código válido. O programa deverá calcular o valor do dígito de controlo e compará-lo com o valor do dígito de controlo do código ISBN13 que foi inserido.

Input

O input tem uma única linha com o código ISBN13 que se pretende validar. A linha tem um número constituído exatamente por 13 algarismos (de 0 a 9).

Output

O output tem uma linha. Se o código ISBN13 for válido, a linha tem a palavra “OK” (sem aspas). No caso contrário, em que o dígito de controlo calculado pelo programa é diferente do dígito de controlo do código que foi introduzido, a linha tem a forma “ERRO *d*” (sem aspas), onde *d* é o valor do dígito de controlo que foi calculado pelo seu programa.

Exemplo 1

Input

9780136091813

Output

OK

Exemplo 2

Input

9780136091817

Output

ERRO 3

Exemplo 3

Input

9789896262180

Output

ERRO 1

Atenção à tensão

Quem se lembra do Sr. Delicado? Ele ofereceu-se prontamente para nos próximos dias mediar uma série de reuniões de negócios entre os representantes de várias potências do mundo empresarial.

Para cada reunião serão convidados os representantes de N empresas (sendo N um número par). O Sr. Delicado disponibilizou, no seu *château*, uma sala com N cadeiras dispostas à volta de uma mesa circular, igualmente espaçadas e numeradas de 1 a N no sentido dos ponteiros do relógio. Em cada cadeira ficará sentado o representante de uma empresa.

Como é natural, há rivalidades entre algumas das empresas, seja por parcerias passadas que não funcionaram da melhor forma, seja por inveja pelo sucesso da concorrência. O Sr. Delicado foi previamente informado se, para cada par de empresas, existe algum tipo de rivalidade entre elas.

Cabe, enfim, ao anfitrião decidir o lugar onde se sentará o representante de cada empresa. É do seu interesse que se conste publicamente que as reuniões no seu *château* decorrem livres de discussões acaloradas e, por isso, quer evitar sentar quaisquer dois rivais frente a frente. Será possível fazê-lo?



Tarefa

Escreva um programa que, tendo em conta as rivalidades que se verificam entre algumas empresas, ajude o Sr. Delicado a descobrir de quantas formas é possível sentar os representantes sem que haja dois rivais frente a frente.

Input

A primeira linha do input contém um inteiro, R , que corresponde ao número de reuniões que terão lugar no *château* do Sr. Delicado durante os próximos dias. Para cada reunião, é fornecida a seguinte informação:

- A primeira linha relativa a uma reunião contém dois inteiros, N e M , respetivamente o número de representantes das empresas que se irão reunir e o número de pares de empresas rivais.
- A segunda linha contém N palavras, separadas por um espaço, indicando os nomes dos representantes das empresas. Cada nome é formado por (de 1 a 20) letras maiúsculas ou minúsculas, sem acentos nem cedilhas. Todos os nomes são diferentes.
- Seguem-se M linhas, cada uma com dois nomes, A e B , separados por um espaço. Tal significa que a empresa cujo representante é A é rival da empresa cujo representante é B . É garantido que ambos os nomes correspondem a dois representantes diferentes e que cada par de rivais não aparece no input mais do que uma vez.

Restrições

$1 \leq R \leq 5$	Número de reuniões
$2 \leq N \leq 12$ e N é par	Número de empresas convidadas para uma reunião
$0 \leq M \leq N \times (N - 1)/2$	Número de pares de empresas rivais numa reunião

Output

O output tem uma linha por cada reunião, pela ordem definida no input. Se, para a respetiva reunião, existir alguma disposição dos representantes das empresas na qual não fiquem dois rivais sentados frente a frente, a linha tem a forma “**Possivel: q** ” (sem aspas nem acento), onde q é o número de formas distintas de sentar os representantes sem que haja dois rivais frente a frente. No caso contrário, a linha tem a palavra “**Impossible**” (sem aspas nem acento).

Exemplo

Input

```
4
4 3
Amadeu Beatriz Carlos Daniel
Amadeu Carlos
Beatriz Carlos
Daniel Beatriz
4 4
Amadeu Beatriz Carlos Daniel
Amadeu Beatriz
Amadeu Carlos
Beatriz Carlos
Daniel Beatriz
4 0
Rita Tiago Manuel Susana
2 1
Paulo Raquel
Paulo Raquel
```

Output

```
Possivel: 8
Impossible
Possivel: 24
Impossible
```

Nota

Dois exemplos de disposições válidas para a primeira reunião, onde o i -ésimo nome corresponde ao representante sentado na i -ésima cadeira ($1 \leq i \leq N$), são:

- Amadeu, Carlos, Beatriz, Daniel;
- Daniel, Amadeu, Carlos, Beatriz.