

## Estradas asfaltadas - o início

O principal dirigente das estradas de Nlogônia está fazendo um levantamento para definir qual deve ser a próxima rodovia a ser construída.

Nlogônia ainda é um país pobre e não existe estrada asfaltada ligando todas as cidades deste país, e por isso é importante traçar uma meta para qual estrada deve ser construída.

Na atual etapa de desenvolvimento do plano estratégico precisamos descobrir quantos núcleos distintos de cidades possuem interconexão. O governo, então, lançou um APP para que os Nlogocienses pudessem ajudar a marcar as cidades que possuem ligação direta por asfalto.

Os Nlogocienses são rápidos e solícitos, e por isso, já preencheram informações a respeito das estradas. Agora é possível determinar quais cidades estão conectadas por estradas.

O dirigente das estradas percebeu uma coisa curiosa, existem alguns núcleos conectados dentro de Nlogônia, ou seja, existe um conjunto de cidades que estão conectados mas outros conjuntos não estão. Veja a figura abaixo:

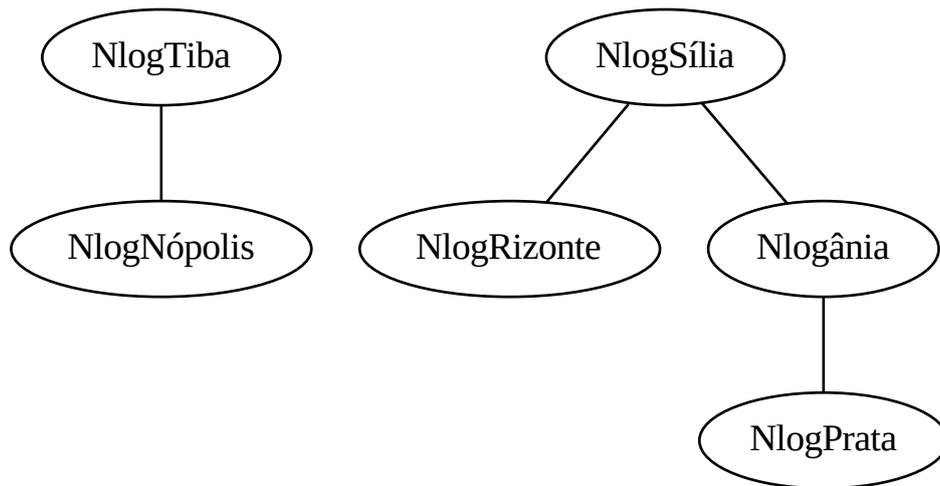


Figure 1: Curiosa Característica de Nlogônia

Na figura acima percebemos dois núcleos de cidades interconectadas. **NlogTiba** e **NlogNópolis** possuem uma estrada asfaltada que se conectam, no entanto nenhuma dessas duas cidades possuem estrada asfaltada para **NlogSília**, **NlogRizonte**, **Nlogânia** ou **NlogPrata**, que por sua vez possuem estradas entre si.

Esse efeito intrigou nosso querido dirigente, e agora ele quer saber quantos núcleos conectados de cidades existem.

O mais interessante é que todas estradas asfaltadas de Nlogônia são vias de mão dupla, então se há uma estrada da cidade A para cidade B, significa que B para A também possui uma estrada (que é a mesma).

Você foi contratado para descobrir essas valiosas informações para nosso dirigente.

## Entrada

A entrada é composta por um único caso de teste. A primeira linha, do caso de teste, possui um número  $V$  ( $1 \leq V < 5000$ ) representando a quantidade de cidades existentes em Nlogônia, as cidades, para simplificar, foram nomeadas de 0 a  $V$ .

Depois da primeira linhas existe um conjunto indeterminado de linhas, com 0 ou mais linhas, terminando em EOF.

As demais linhas possuem dois inteiros  $v$  e  $w$  ( $0 \leq v, w \leq V$ ) informando a existência de uma estrada conectando as cidades  $v$  para  $w$ . Você pode considerar que também existe uma estrada de  $w$  para  $v$ . É garantido que a estrada será passada uma única vez, ou seja, se existir na entrada  $vw$ , não será passado  $wv$ .

## Saída

A saída é composta por uma única linha contendo um inteiro que representa quantidade de núcleos de cidades conectadas.

## Exemplos

### Exemplo de entrada

1

### Saída para o exemplo acima

1

### Exemplo de entrada

6  
0 1  
2 3  
2 4  
4 5

### Saída para o exemplo acima

2

### Exemplo de entrada

13  
10 7  
10 6  
10 2  
7 2  
6 12  
12 5  
12 3  
12 2  
11 8  
9 0  
9 1  
1 0  
1 4

### Saída para o exemplo acima

3

### Exemplo de entrada

3  
1 2

### Saída para o exemplo acima

2

### Exemplo de entrada

10  
1 9  
9 8  
8 7

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

**Saída para o exemplo acima**

1

**Exemplo de entrada**

5000  
1 4999  
4999 5

**Saída para o exemplo acima**

4998

`\textit{\rightline{Author: Bruno Ribas }}`

# Natureza

Na natureza, existem as cadeias alimentares. Na base dessas cadeias, geralmente temos os vegetais. Pequenos animais comem esses vegetais e animais maiores comem os menores. Podem ocorrer ciclos em uma cadeia, como quando um animal morre e inicia-se um processo de decomposição que transformará seu corpo em minerais que são uma fonte de energia para os vegetais.

Neste problema você deverá encontrar a maior cadeia alimentar para um dado grupo de criaturas. Você pode considerar que se A é o predador de B então eles estão na mesma cadeia.

## Entrada

O arquivo de entrada contém vários conjuntos de teste. A descrição de cada conjunto é dada a seguir:

Cada conjunto começa com dois inteiros  $C$  ( $1 \leq C \leq 5000$ ), o número de criaturas, e  $R$  ( $0 \leq R \leq 5000$ ), o número de relações. Seguem  $C$  linhas com os nomes das criaturas, cada nome é formado somente por letras minúsculas (a, b, ..., z). Nenhum nome possui mais do que 30 caracteres. Por fim, haverá  $R$  linhas descrevendo as relações. Cada linha terá o nome de 2 criaturas, indicando que a segunda criatura é um predador da primeira.

Você pode assumir que nenhuma criatura é predadora dela mesma.

A entrada é terminada por um conjunto onde  $C = R = 0$ . Este conjunto não deve ser processado. Há uma linha em branco entre dois conjuntos de entrada.

## Saída

Para cada conjunto de entrada produza uma linha de saída, o tamanho da maior cadeia alimentar.

## Exemplo

### Entrada:

```
5 2
caterpillar
bird
horse
elefant
herb
herb caterpillar
caterpillar bird
```

```
0 0
```

### Saída:

```
3
```

# Estradas asfaltadas - viagem aérea

O ano não está fácil para Nlogônia. Em meio planejamento de construção das rodovias para conectar as cidades, a natureza resolveu castigar e destruiu TODAS estradas de terra deste nosso querido país.

Vimos anteriormente (no exercício anterior) que o país possui algumas componentes conexas, ou seja, núcleos de cidades conectadas entre si por estradas asfaltadas, mas sem conexão geral com as demais cidades do país. Veja abaixo a característica da cidade:

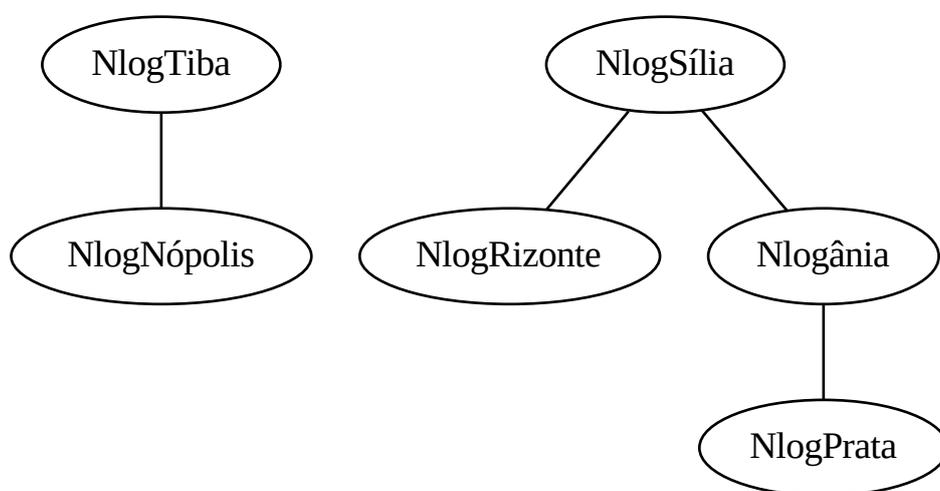


Figure 1: Estradas asfaltadas de Nlogônia

João é um jovem vendedor de **ponteiros** e está arrasado com a triste chuva que tomou conta de Nlogônia. Mais triste ainda por ter que viajar para algumas cidades. O dinheiro de João é “curto” e por isso ele tenta viajar com o bom e velho ônibus sempre que possível.

O problema é que João tem dinheiro para apenas uma passagem de avião (todos os vôos possuem o mesmo preço) e ele precisa ir para a região que possua o maior número de cidades conectadas por estrada.

João clama por ajuda!

## Entrada

A entrada é composta por um único caso de teste. A primeira linha, do caso de teste, possui dois números inteiros  $V$  ( $1 \leq V < 5000$ ) e  $J$  ( $0 \leq J < V$ ) representando, respectivamente, a quantidade de cidades existentes em Nlogônia, as cidades, para simplificar, foram nomeadas de 0 a  $V$ , e a cidade em que João se encontra.

Depois da primeira linhas existe um conjunto indeterminado de linhas, com 0 ou mais linhas, terminando em EOF.

As demais linhas possuem dois inteiros  $v$  e  $w$  ( $0 \leq v, w \leq V$ ) informando a existência de uma estrada conectando as cidades  $v$  para  $w$ . Você pode considerar que também existe uma estrada de  $w$  para  $v$ . É garantido que a estrada será passada uma única vez, ou seja, se existir na entrada  $vw$ , não será passado  $wv$ .

## Saída

A saída é composta por uma única linha, e dependendo do caso pode variar:

- Se João já estiver na maior componente conexa seu programa deve imprimir: **Bora pra estrada**
- Se João estiver na maior componente conexa e ela possuir grau 0 (sem aresta), você deve imprimir: **Fique em casa**
- Caso exista uma componente conexa maior que a de João vá para o vértice de menor índice, imprimindo: **Vamos para XX**, sendo *XX* o número do vértice
- Em caso de empate nos tamanhos das componentes prefira a seguinte ordem:
  - Se for do mesmo tamanho que a que a João já está, fique na mesma componente e imprima **Bora pra estrada**
  - Caso contrário prefira a componente que possua a cidade com menor índice

## Exemplos

### Exemplo de entrada

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

### Saída para o exemplo acima

Vamos para 2

### Exemplo de entrada

```
13 0
10 7
10 6
10 2
7 2
6 12
12 5
12 3
12 2
11 8
9 0
9 1
1 0
1 4
```

### Saída para o exemplo acima

Vamos para 2

### Exemplo de entrada

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

### Saída para o exemplo acima

Bora pra estrada

### Exemplo de entrada

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

### Saída para o exemplo acima

Bora pra estrada

### Exemplo de entrada

13 8  
10 7  
10 6  
10 2  
7 2  
6 12  
12 5  
12 3  
12 2  
11 8  
9 0  
9 1  
1 0  
1 4

### Saída para o exemplo acima

Vamos para 2

### Exemplo de entrada

7 5  
0 6  
0 3  
1 5  
5 2

### Saída para o exemplo acima

Bora pra estrada

### Exemplo de entrada

13 6  
10 7  
10 6  
10 2  
7 2  
6 12  
12 5  
12 3  
12 2  
11 8  
9 0  
9 1  
1 0  
1 4

### Saída para o exemplo acima

Bora pra estrada

### Exemplo de entrada

7 4  
0 6  
0 3  
1 5  
5 2

### Saída para o exemplo acima

Vamos para 0

**Exemplo de entrada**

1 0

**Saída para o exemplo acima**

Fique em casa

**Exemplo de entrada**

1000 7  
0 999  
5 999  
6 999  
7 999  
8 999  
9 999

**Saída para o exemplo acima**

Bora pra estrada

**Exemplo de entrada**

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

**Saída para o exemplo acima**

Bora pra estrada

`\textit{\rightline{Author: Bruno Ribas }}`

# Cai fora LAG

Manter uma rede de internet de fibra em funcionamento é uma tarefa muito difícil. Diversos fatores afetam o tempo de resposta de comunicação da rede e cada vez mais fica difícil descobrir o melhor caminho que um pacote pode percorrer, ainda mais que os switches podem fazer parte de empresas diferentes, e os custos de passar um pacote seu por ela pode custar mais caro.

Por isso que na vida real as operadoras tentam não usar a rede das concorrentes para trafegar pacotes. Por exemplo, se você usa internet da NET/CLARO e vai se conectar ao youtube, a sua conexão nunca vai passar por roteadores da VIVO. Pois a VIVO cobraria uma taxa de tráfego de empresa externa, quase que como um pedágio de pacotes.

Para resolver este problema, uma notória equipe de desenvolvedores criou o *CAI FORA LAG*, uma ferramenta que entende como a rede funciona e sempre descobre o menor caminho (e mais barato) para trafegar os pacotes, isso a qualquer momento da rede.

Para garantir uma economia maior, a *CAI FORA LAG* ainda instalou um sistema chamado **SUPER CONEXÃO**, que nada mais é que um super switch (muito caro) com um link muito grande que conecta dois pontos de conexão a uma latência baixa e a um custo muuuito baixo, garantindo assim, uma economia nos gastos da empresa e uma melhora no desempenho da rede dos clientes (essencialmente serve apenas para a *CAI FORA LAG* economizar dinheiro ao ficar roteando os pacotes dos clientes).

O pessoal do *CAI FORA LAG* precisa encontrar mais parceiros para conseguir colocar em prática a instalação da **SUPER CONEXÃO**. E, por isso, precisa saber quanto está economizando ao utilizar este método em lugares específicos.

No desenho abaixo temos uma representação de um pedaço da rede, bem simples, que mostra que as conexões entre os nós da rede de fibra (note que não há direção, pois a rede vai e volta) com um custo associado nas arestas, percebemos que o caminho direto do vértice 0 para 1 é de 8.

Temos também, na figura, uma rede de **SUPER CONEXÃO** ligando os vértices 3 para 1 a um custo de  $-2$ , note que esta conexão é direcionada, ou seja, pacotes podem ser enviados de 3 para 1 diretamente, mas o caminho inverso não ocorre.

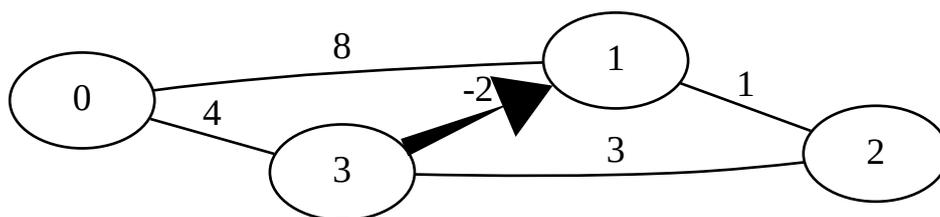


Figure 1: Pequeno exemplo de conexão

Então, no desenho acima, se eu quiser enviar um pacote do vértice 0 para o vértice 2, o custo sem **SUPER CONEXÃO** é 7, passando por 0 - 3 - 2 e quando se considera a existência da **SUPER CONEXÃO** o custo passa a ser 3 pois o pacote seguiria o caminho 0 - 3 - 1 - 2.

Enviar um pacote de 1 para 0 o custo é o mesmo sem e com a **SUPER CONEXÃO**, pois não há vantagens em se utilizar alguma rota que passe pela **SUPER CONEXÃO**.

A sua tarefa é descobrir o menor custo de operação de envio de pacotes de uma origem a um destino sem considerar a existência dos nós **SUPER CONEXÃO** e depois considerando a existência deles.

## Entrada

A entrada é composta por um único caso de teste. A primeira linha, do caso de teste, possui um número inteiro  $V$  ( $2 \leq V \leq 2000$ ) representando a quantidade de switches (vértices) que foram nomeadas de 0 a  $V$ .

A partir da segunda linha, cada linha é composta por quatro inteiros  $v, w, c$  e  $t$  ( $0 \leq v, w < V, -1000 \leq c \leq 1000, 0 \leq t \leq 1$ ), com  $v$  e  $w$  informando a conexão de fibra entre  $v$  e  $w$ ,  $c$  sendo o custo de se utilizar este caminho e  $t$  representando o tipo da conexão 0 para bidirecional e 1 para **SUPER CONEXÃO** que é direcional (não existindo o caminho  $w v$  com mesmo custo).

Quando  $v = w = c = t = 0$ , representa que o fim da descrição das conexões e iniciam-se as perguntas da rede.

Para as perguntas cada linha é composta por dois inteiros  $o$  e  $x$  ( $0 \leq o, x < V$ ) representando a pergunta de qual é o custo para enviar um pacote de  $o$  para  $x$ .

A entrada termina em EOF.

É garantido que ao menos uma aresta e uma pergunta faça parte da entrada.

## Saída

Para cada pergunta você deve imprimir uma única linha contendo:

- Dois inteiros  $X$  e  $S$ , sendo  $X$  o custo de enviar pacotes de  $o$  para  $x$  SEM utilizar a super conexão e  $S$  sendo o custo de se enviar pacotes utilizando a **SUPER CONEXÃO**
- **Impossibru**, quando for impossível calcular o custo com ou sem a **SUPER CONEXÃO**

## Exemplos

### Exemplo de Entrada

```
4
0 1 8 0
0 3 4 0
1 2 1 0
3 2 3 0
3 1 -2 1
0 0 0 0
0 2
1 0
2 1
```

### Saída para o exemplo acima

```
7 3
8 8
1 1
```

### Exemplo de Entrada

```
4
0 1 8 0
0 3 4 0
1 2 1 0
3 2 3 0
3 1 -2 1
2 0 -5 1
0 0 0 0
0 2
1 0
2 1
```

### Saída para o exemplo acima

```
Impossibru
Impossibru
Impossibru
```

### Exemplo de Entrada

```
4
0 1 8 0
0 3 4 0
1 2 1 0
3 2 3 0
3 1 -2 1
0 2 -5 1
0 0 0 0
0 2
1 0
2 1
```

### Saída para o exemplo acima

```
7 -5
8 8
1 1
```

### Exemplo de Entrada

```
5
0 1 8 0
0 3 4 0
1 2 1 0
3 2 3 0
3 1 -2 1
0 0 0 0
0 2
0 4
```

### Saída para o exemplo acima

```
7 3
Impossibru
```

### Exemplo de Entrada

```
10
6 3 820 0
6 1 270 0
1 0 160 0
9 0 489 0
3 4 574 0
2 1 28 0
8 9 368 0
8 7 138 0
7 5 30 0
2 8 313 0
6 7 153 0
7 1 272 0
4 7 182 0
9 2 369 0
9 5 80 0
5 8 877 0
2 6 689 0
7 9 219 0
8 1 334 0
4 2 453 0
2 3 86 0
0 5 239 0
7 0 245 0
5 2 314 0
3 7 324 0
4 8 164 0
```

8 6 897 0  
5 3 371 0  
2 0 280 0  
0 0 0 0  
0 1  
0 2  
0 3  
0 4  
0 5  
0 6  
0 7  
0 8  
0 9  
1 0  
1 2  
1 3  
1 4  
1 5  
1 6  
1 7  
1 8  
1 9  
2 0  
2 1  
2 3  
2 4  
2 5  
2 6  
2 7  
2 8  
2 9  
3 0  
3 1  
3 2  
3 4  
3 5  
3 6  
3 7  
3 8  
3 9  
4 0  
4 1  
4 2  
4 3  
4 5  
4 6  
4 7  
4 8  
4 9  
5 0  
5 1  
5 2  
5 3  
5 4  
5 6  
5 7  
5 8  
5 9  
6 0  
6 1  
6 2  
6 3  
6 4  
6 5

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

**Saída para o exemplo acima**

160 160  
188 188  
274 274  
427 427  
239 239  
398 398  
245 245  
383 383  
319 319  
160 160  
28 28  
114 114  
454 454  
302 302  
270 270  
272 272  
334 334  
382 382  
188 188  
28 28  
86 86  
453 453  
314 314  
298 298  
300 300  
313 313  
369 369  
274 274  
114 114  
86 86  
506 506

354 354  
384 384  
324 324  
399 399  
434 434  
427 427  
454 454  
453 453  
506 506  
212 212  
335 335  
182 182  
164 164  
292 292  
239 239  
302 302  
314 314  
354 354  
212 212  
183 183  
30 30  
168 168  
80 80  
398 398  
270 270  
298 298  
384 384  
335 335  
183 183  
153 153  
291 291  
263 263  
245 245  
272 272  
300 300  
324 324  
182 182  
30 30  
153 153  
138 138  
110 110  
383 383  
334 334  
313 313  
399 399  
164 164  
168 168  
291 291  
138 138  
248 248  
319 319  
382 382  
369 369  
434 434  
292 292  
80 80  
263 263  
110 110  
248 248

### Exemplo de Entrada

18  
15 7 -938 1

11 0 600 0  
13 14 587 0  
0 15 274 0  
9 6 885 0  
12 17 925 0  
10 13 40 0  
1 5 49 0  
13 4 247 0  
5 6 78 0  
11 12 84 0  
1 15 742 0  
12 4 962 0  
4 9 528 0  
17 16 14 0  
8 5 120 0  
11 1 943 0  
7 14 767 0  
2 17 673 0  
2 14 399 0  
11 17 923 0  
13 6 587 0  
10 5 348 0  
0 2 557 0  
11 13 793 0  
7 3 59 0  
6 15 660 0  
11 16 596 0  
7 6 39 0  
14 15 0 0  
7 10 87 0  
7 8 651 0  
14 11 230 0  
9 11 944 0  
17 15 369 0  
12 0 98 0  
13 2 956 0  
4 16 839 0  
13 3 940 0  
4 14 222 0  
1 14 784 0  
15 2 548 0  
7 4 869 0  
16 3 263 0  
11 2 174 0  
3 4 582 0  
17 8 140 0  
15 16 380 0  
12 13 582 0  
0 1 950 0  
1 13 566 0  
8 10 37 0  
10 9 492 0  
14 10 316 0  
13 0 327 0  
7 17 673 0  
11 4 263 0  
7 13 19 0  
16 6 994 0  
16 2 707 0  
17 6 310 0  
16 8 529 0  
12 3 659 0  
0 0 0 0  
0 1

0 2  
0 3  
0 4  
0 5  
0 6  
0 7  
0 8  
0 9  
0 10  
0 11  
0 12  
0 13  
0 14  
0 15  
0 16  
0 17  
1 0  
1 2  
1 3  
1 4  
1 5  
1 6  
1 7  
1 8  
1 9  
1 10  
1 11  
1 12  
1 13  
1 14  
1 15  
1 16  
1 17  
2 0  
2 1  
2 3  
2 4  
2 5  
2 6  
2 7  
2 8  
2 9  
2 10  
2 11  
2 12  
2 13  
2 14  
2 15  
2 16  
2 17  
3 0  
3 1  
3 2  
3 4  
3 5  
3 6  
3 7  
3 8  
3 9  
3 10  
3 11  
3 12  
3 13  
3 14

3 15  
3 16  
3 17  
4 0  
4 1  
4 2  
4 3  
4 5  
4 6  
4 7  
4 8  
4 9  
4 10  
4 11  
4 12  
4 13  
4 14  
4 15  
4 16  
4 17  
5 0  
5 1  
5 2  
5 3  
5 4  
5 6  
5 7  
5 8  
5 9  
5 10  
5 11  
5 12  
5 13  
5 14  
5 15  
5 16  
5 17  
6 0  
6 1  
6 2  
6 3  
6 4  
6 5  
6 7  
6 8  
6 9  
6 10  
6 11  
6 12  
6 13  
6 14  
6 15  
6 16  
6 17  
7 0  
7 1  
7 2  
7 3  
7 4  
7 5  
7 6  
7 8  
7 9  
7 10

7 11  
7 12  
7 13  
7 14  
7 15  
7 16  
7 17  
8 0  
8 1  
8 2  
8 3  
8 4  
8 5  
8 6  
8 7  
8 9  
8 10  
8 11  
8 12  
8 13  
8 14  
8 15  
8 16  
8 17  
9 0  
9 1  
9 2  
9 3  
9 4  
9 5  
9 6  
9 7  
9 8  
9 10  
9 11  
9 12  
9 13  
9 14  
9 15  
9 16  
9 17  
10 0  
10 1  
10 2  
10 3  
10 4  
10 5  
10 6  
10 7  
10 8  
10 9  
10 11  
10 12  
10 13  
10 14  
10 15  
10 16  
10 17  
11 0  
11 1  
11 2  
11 3  
11 4  
11 5

11 6  
11 7  
11 8  
11 9  
11 10  
11 12  
11 13  
11 14  
11 15  
11 16  
11 17  
12 0  
12 1  
12 2  
12 3  
12 4  
12 5  
12 6  
12 7  
12 8  
12 9  
12 10  
12 11  
12 13  
12 14  
12 15  
12 16  
12 17  
13 0  
13 1  
13 2  
13 3  
13 4  
13 5  
13 6  
13 7  
13 8  
13 9  
13 10  
13 11  
13 12  
13 14  
13 15  
13 16  
13 17  
14 0  
14 1  
14 2  
14 3  
14 4  
14 5  
14 6  
14 7  
14 8  
14 9  
14 10  
14 11  
14 12  
14 13  
14 15  
14 16  
14 17  
15 0  
15 1

15 2  
15 3  
15 4  
15 5  
15 6  
15 7  
15 8  
15 9  
15 10  
15 11  
15 12  
15 13  
15 14  
15 16  
15 17  
16 0  
16 1  
16 2  
16 3  
16 4  
16 5  
16 6  
16 7  
16 8  
16 9  
16 10  
16 11  
16 12  
16 13  
16 14  
16 15  
16 17  
17 0  
17 1  
17 2  
17 3  
17 4  
17 5  
17 6  
17 7  
17 8  
17 9  
17 10  
17 11  
17 12  
17 13  
17 14  
17 15  
17 16

**Saída para o exemplo acima**

Impossibru  
Impossibru









