

# Caderno de Provas – *Original*

1<sup>a</sup> Seletiva Interna – 2019/1

UDESC–Joinville – IFMS – UTFPR

Servidor BOCA (Arena Joinville):

<http://200.19.107.69/>



## Organização e Realização:

Karina Girardi Roggia (coordenação geral), Peter Laureano Brendel (coordenação técnica),  
Vinicius Gasparini (coordenação técnica)

Criação de problemas: Adilson Jonck Junior, Felipe Weiss, Felipe Marchi, Claudio Cesar de Sá,  
Gabriel Hermann Negri, Lucas Hermann Negri.

Patrocinador 2019: Neogrid

## Lembretes:

- Aos *javaneiros*: o nome da classe deve ser o mesmo nome do arquivo a ser submetido.  
Ex: classe **petrus**, nome do arquivo **petrus.java**;
- Exemplo de leitura de entradas que funcionam:

```
Java: (import java.util.Scanner)
Scanner in = new Scanner(System.in);
ou
Scanner stdin = new Scanner(new BufferedReader(new InputStreamReader(System.in)));
```

```
C: (#include <stdio.h>)
int integer1; scanf("%d", &integer1);
```

```
C++: (#include <iostream>)
int integer1; std::cin >> integer1;
```

Exemplo de saída de entradas:

```
Java: System.out.format("%d %d\n", integer1, integer2);
C: printf("%d %d\n", integer1, integer2);
C++: std::cout << integer1 << " " << integer2 << std::endl;
```

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova;
- A correção é automatizada, portanto,  **siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos**. Deve-se considerar entradas e saídas padrão;
- Todos os compiladores (Java, Python, C e C++) são padrões da distribuição Ubuntu versão 16.04 (gcc C11 habilitado);
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos;
- Utilize o *clarification* para dúvidas da prova. Os juízes podem **opcionalmente** atendê-lo com respostas acessíveis a todos;
- Algumas interfaces estão disponíveis nas máquinas Linux, que podem ser utilizada no lugar da *Unity*. Para isto, basta dar *logout*, e selecionar a interface desejada. Usuário e senha: *udesc*;
- Ao pessoal local: **cuidado com os pés sobre as mesas para não desligarem nenhum estabilizador/computador de outras equipes!**
- Na sequência do *Originario*, segue a *Original*;

## Patrocinador e Agradecimentos

- Neogrid – Patrocinador oficial do ano de 2019;
- DCC/UEDESC;
- Aos bolsistas deste ano pelo empenho;
- Alguns, muitos outros anônimos.

# Original

1ª Seletiva Interna da UDESC IFMS e UTFPR – 2019

27 de agosto de 2019

## Conteúdo

1	Problema A: JoJo	4
2	Problema B: Encurralado	6
3	Problema C: Intervalo	8
4	Problema D: A Pedreira no Tromba	10
5	Problema E: Bó Aposta	12
6	Problema F: Faltou o Título Weiss	14
7	Problema G: Gado	16
8	Problema H: Kangaroo Word	18
9	Problema I: Cracker	20
10	Problema J: Tamanho da Janela	21
11	Problema K: Dois malucos	22

---

Atenção quanto aos nomes e números dos problemas!!!

---

# 1 Problema A: JoJo

Arquivo: A.[c|cpp|java|py]

Tempo limite: 2s

Faustinho e Olokinho são dois irmãos muito competitivos, e durante a páscoa seus pais deram para eles dois grafos não direcionados idênticos. Faustinho, sendo o irmão mais velho, decidiu transformar seu grafo em uma árvore mínima, ou seja, um grafo onde:

1. É possível de um vértice alcançar qualquer outro
2. Existe um caminho único para cada par de vértices
3. A soma do custo das arestas na árvore resultante é mínima

## Problema

Olokinho gostou da ideia, mas não gostaria de imitar seu irmão mais velho, então por ser o segundo filho decidiu transformar seu grafo na segunda árvore mínima, ou seja, uma árvore que tenha um custo diferente da árvore mínima e que seja com o menor custo possível. Olokinho sabe como achar a árvore mínima, porém não tem ideia de como achar a segunda árvore mínima e por isso decidiu pedir sua ajuda.

É garantido que Faustinho consegue criar uma árvore mínima partindo de seu grafo inicial.

## Entrada

A primeira linha da entrada possui dois inteiros  $N$  e  $M$  referentes ao número de nodos do grafo inicial e o número de arestas do grafo inicial. As próximas  $M$  linhas contém três inteiros  $U, V, W$  indicando que existe uma aresta entre os nodos  $U$  e  $V$  de peso  $W$ .

## Restrições

- $1 \leq N, Q \leq 10^5$
- $N - 1 \leq M \leq \min(10^5, \frac{N * (N - 1)}{2})$
- $1 \leq U, V \leq N$
- $1 \leq W \leq 10^5$

## Saída

O custo total da segunda árvore mínima que Olokinho deseja ou  $-1$  caso não exista a segunda árvore mínima com custo diferente da árvore mínima.

Exemplo de Entrada	Exemplo de Saída
5 7 1 2 7 2 5 2 4 3 1 2 3 3 5 3 1 2 4 4 2 5 5	12
3 3 1 2 1 2 3 1 1 3 1	-1

## 2 Problema B: Encurralado

Arquivo: B.[c|cpp|java|py]

Tempo limite: 2s

Menino Dilsinho é um estudante que mora na infame cidade de Matrizville, e enquanto estava em casa se arrumando para ir para a faculdade, começou a chover! Dilsinho é um menino sabido dos caminhos de Matrizville, e possui uma rota fixa que utiliza durante os dias ensolarados na cidade (são poucos), porém quando chove, algumas partes de terra podem ficar alagadas.

### Problema

A cidade de Matrizville pode ser definida como uma matriz de tamanho  $N \times N$ , onde a casa de Dilsinho fica localizada na posição  $(1, 1)$  e sua universidade (UDEM\*\*) se localiza na posição  $(N, N)$ . Ele pode se locomover para qualquer pedaço de terra firme adjacente a sua localização atual (Cima, baixo, esquerda, direita). É garantido que a casa de Dilsinho e sua universidade não foram alagados pela chuva.

Dilsinho gostaria de saber se existe um jeito de chegar em sua universidade partindo de sua casa e, conhecendo suas habilidades como programador, decidiu pedir a sua ajuda!

### Entrada

A entrada é dada por um inteiro  $N$ , referente ao tamanho de Matrizville. Seguem  $N$  linhas, cada uma delas contendo  $N$  inteiros  $N_i$ .

### Restrições

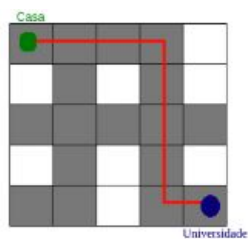
- $1 \leq N \leq 1000$
- Considere um pedaço de terra firme se  $N_i = 1$ , caso contrário a terra está alagada.

### Saída

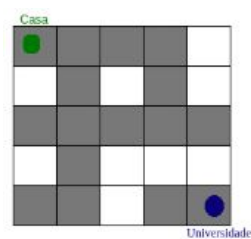
Se existir um caminho saindo de  $(1, 1)$  e chegando em  $(N, N)$ , imprima “Vai pra aula, Dilsinho!” (sem aspas). Caso contrário, imprima “Fica em casa, Dilsinho!” (sem aspas).

Exemplo de Entrada	Exemplo de Saída
<pre> 5 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 0 1 0 1 1 0 1 1 </pre>	Vai pra aula, Dilsinho!
<pre> 5 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 0 1 1 0 1 1 </pre>	Fica em casa, Dilsinho!

\*\*Universidade Dos Empecilhos de Matrizville



(a) Primeira Entrada



(b) Segunda Entrada

### 3 Problema C: Intervalo

Arquivo: C.[c|cpp|java|py]

Tempo limite: 2s

Em uma noite qualquer com seus amigos, Manoel decidiu pedir uma pizza para comer. Ao chegar a pizza, eles começaram a dividi-la e enquanto Manoel estava dividindo os pedaços da pizza entre os seus amigos ele pensou no seguinte problema:

Se eu tenho um valor  $X$ , qual é a soma de todos os divisores de  $X$ ? Manoel, por ser um garoto habilidoso com números conseguiu decifrar rapidamente essa charada, então decidiu dificultá-la:

#### Problema

Se eu tenho um valor  $X$  e  $Y$ , qual é o resto da divisão por  $K$  da soma dos divisores de todos os números entre o intervalo de  $X$  até  $Y$  inclusos?

Manoel está comendo pizza agora, logo, você deve resolver essa nova charada!

#### Entrada

A primeira linha contém dois inteiros  $N$ ,  $Q$  referentes ao valor máximo de  $X$  e  $Y$  e o número de consultas que você deve responder, respectivamente.

Nas próximas  $Q$  linhas seguem três inteiros  $X$ ,  $Y$ ,  $K$ , referente a soma dos divisores de todos os números entre o intervalo de  $X$  até  $Y$  (inclusos) e o valor  $K$  referente ao resto da divisão da soma total.

#### Restrições

- $1 \leq N, Q \leq 2 * 10^5$
- $1 \leq K \leq 10^9 + 7$
- $1 \leq X \leq Y \leq N$

#### Saída

Para cada consulta, imprima o valor correspondente ao resto da divisão da soma de todos os divisores de todos os números entre  $X$  e  $Y$  por  $K$ .

Exemplo de Entrada	Exemplo de Saída
10 4	21
1 5 100	3
2 2 10	3
1 10 7	0
4 7 33	

Para este problema, temos que considerar o intervalo  $[1, 10]$ . Na primeira consulta, o intervalo requerido é de 1 até 5, ou seja, temos:



- $1 \rightarrow [1] = 1$
- $2 \rightarrow [1, 2] = 3$
- $3 \rightarrow [1, 3] = 4$
- $4 \rightarrow [1, 2, 4] = 7$
- $5 \rightarrow [1, 5] = 6$
- resposta:  $21 \% 100 = 21$

## 4 Problema D: A Pedreira no Tromba

Arquivo: D.[c|cpp|java|py]

Tempo limite: 2s

Estando no corredor do 3o. piso do DCC/UDESC, avista-se a oeste, a Serra do Mar. A Serra do Mar é avistada de quase toda Joinville, e o por do sol é um momento belo de se apreciar. Da região, o Morro da Tromba é o mais imponente, pelos seus 967 metros e a daqui proximidade da cidade. Uma região exuberante de mata nativa, verde oliva, água corrente, etc. Contudo, há alguns anos ocorreu a exploração de uma pedreira naquele morro. Este fato, vem mudando a sua paisagem significativamente. A face sul, tem sido arrasada por esta depredação neste morro, pela remoção de suas pedras para uso na construção civil, pavimentação, etc. Ou seja, todo morro é formado por grandes rochas e pedras. Esta exploração é feita por explosivos na fachada do morro. Para isto, há necessidade de se subir a pedreira até o topo, ora descer até a base. As pedras que aparecem são muito grandes e uma pessoa normal caminha apenas sobre as pedras pequenas. Assim, os peritos em explosivos ficam atravessando toda esta pedreira, procurando o caminho que menos lhe dêem desgaste. Ou seja, eles escolhem as menores pedras, mas que nem sempre leva ao menor caminho da base ao topo, e vice-versa. Assim, estas pedras terão um valor numérico que representam o grau de uma pessoa caminhar sobre ela. A pedreira será representada por uma matriz de valores, e um caminho será definido entre algum e único ponto da primeira linha (topo da pedreira), há um único ponto na última linha (a base).

Exemplificando: seja uma matriz  $6 \times 6$  que represente as dimensões da pedreira e os valores das pedras:

```
1  2   99  111 112 10
99 111 2   3   2   5
1  22  3 1111 2222 99
2 2222 222 100 2   99
4  2   13  2   7   9
99 99 999 999 99  9
```

O percurso com menor custo, iniciando em algum ponto da primeira linha é ilustrado pela figura abaixo por \*:

```
# # # # # *
# # * * * *
* * * # # #
* # # # # #
* * * * * *
# # # # # *
```

Assim, o ponto de partida foi encontrado como sendo (1,6) e a chegada no ponto (6,6). O custo mínimo deste caminho é 96.

### Problema

Dada uma matriz  $M \times N$ , encontre o custo do menor caminho de algum ponto da primeira linha até algum ponto da última linha. Os pontos da primeira e da última linha são únicos nesta linha!

## Entrada

A primeira linha de entrada contém um único inteiro positivo  $C$  indicando o número de casos de testes. Na linha seguinte virá a dimensão da pedreira, dada por dois números inteiros positivos  $M$  e  $N$ . Nas  $M$  linhas seguintes contém os valores inteiros desta matrix  $M \times N$ . Acompanhe os exemplos de entrada e saída.

## Restrições

- $1 < C \leq 100$
- $5 \leq M, N \leq 50$

## Saída

Para cada caso de teste, imprima como saída, no seguinte formato:

Caso #C : CUSTO: e o valor do caminho mínimo. Ao final de cada saída, salte de linha. Acompanhe os exemplos de entrada e saída.

Exemplos de Entrada	Exemplos de Saída
3 5 4 1 1 99 99 99 1 1 1 1 99 99 1 1 1 1 1 1 99 99 99 6 6 1 2 99 111 112 10 99 111 2 3 2 5 1 22 3 1111 2222 99 2 2222 222 100 2 99 4 2 13 2 7 9 99 99 999 999 99 9 5 4 1 21 99 99 99 1 2 3 1 99 99 4 99 1 2 1 99 99 1 99	Caso #1 : CUSTO: 10 Caso #2 : CUSTO: 96 Caso #3 : CUSTO: 35

PS: a formulação deste problema é fictício, mas, a depredação ecológica-ambiental é real!

## 5 Problema E: Bó Aposta

Arquivo: E.[c|cpp|java|py]

Tempo limite: 2s

O menino Cláudio é um grande fã de competições de nado em águas abertas, viajando por todo o Brasil (de Foz do Iguaçu até Salvador!) para poder assistir essas competições. Cláudio é um garoto muito peculiar, e adora tentar adivinhar como será o ranking final da competição com os seus amigos, sendo que em diversas ocasiões já ganhou muito dinheiro deles em apostas sobre suas predições!

### Problema

Cláudio está assistindo agora uma competição com  $N$  participantes, junto de seus amigos Roger e Mário, quando Mário decide fazer uma aposta com Cláudio. Mário aposta que Cláudio não conseguirá prever quem ficará nas primeiras  $K$  posições do ranking. Cláudio, ouvindo isso, começou a pensar se deveria ou não aceitar a aposta e, depois de um tempo decidiu o seguinte: Se o número de maneiras diferentes de se obter as  $K$  primeiras posições for menor ou igual que um valor  $X$ , então Cláudio ficará confiante de suas habilidades e irá apostar. Caso contrário, Cláudio ficará intimidado com o número de possibilidades e pagará uma pizza para seus amigos.

### Entrada

A entrada é dada por uma linha contendo três inteiros  $N, K, X$ , onde  $N$  é o número de participantes da competição,  $K$  é o tamanho do ranking que Cláudio deverá tentar adivinhar e  $X$  é o seu valor de decisão.

### Restrições

- $1 \leq N \leq 12$
- $1 \leq K \leq N$
- $1 \leq X \leq 10^9$

### Saída

A saída consistirá da frase “Apostar!” caso Cláudio esteja confiante em aceitar a aposta e “Pizza!” caso decida que não está confiante o suficiente para apostar.

Atenção: Imprima as frases exatamente como descritas na saída padrão.

Exemplo de Entrada	Exemplo de Saída
3 3 6	Apostar!
5 3 10	Pizza!
10 1 1000	Apostar!

No primeiro caso, a quantidade de formas distintas de se montar as 3 primeiras colocações é igual 6 (ABC, ACB, BAC, BCA, CAB, CBA), logo Cláudio está confiante em apostar. (Note que “A”, “B”, “C”, se tratam dos nadadores distintos e suas colocações respectivas) No segundo caso, a quantidade de formas distintas de se formar as 3 primeiras colocações é igual a 60, então Cláudio prefere não apostar. No último caso, a quantidade de maneiras distintas de se formar a primeira colocação é igual ao número de competidores, que é 10, portanto o garoto Cláudio irá apostar.

## 6 Problema F: Faltou o Título Weiss

Arquivo: F.[c|cpp|java|py]

Tempo limite: 2s

A princesa Carolina adora aventuras e, na sua expedição atual ela encontrou  $N$  salas de tesouros, onde a  $i$ -ésima sala contém uma quantidade  $C_i$  de moedas de ouro! Como a princesa Carolina é uma pessoa muito ambiciosa, ela tenta obter todos os tesouros para si. Porém, ao tentar fazê-lo, uma maga aparece dizendo o seguinte: “Somente aqueles que passarem pelo meu árduo teste terão o direito de tocar neste tesouro!”. O teste se resume a  $Q$  perguntas, em cada pergunta é fornecido um intervalo  $(i,j)$ . Para cada pergunta, deve-se dizer a soma total de tesouros que não estão entre o intervalo  $(i,j)$ .

### Problema

Carolina pediu a sua ajuda para passar neste árduo teste da maga, e Como Carolina é também uma princesa muito generosa, ela irá lhe recompensar com uma pizza!

### Entrada

A primeira linha da entrada possui 2 inteiros  $N$  e  $Q$ , referentes ao número de salas de tesouro e a quantidade de perguntas no teste árduo.

A segunda linha possui  $N$  inteiros separados, referentes aos valores  $C_1, C_2, \dots, C_N$ , ou seja, a quantidade de moeda em cada sala de tesouro. Seguem-se  $Q$  linhas, cada uma delas contendo 2 valores  $i$  e  $j$ , referentes ao intervalo da pergunta.

### Restrições

- $1 \leq N \leq 10^6$
- $1 \leq Q \leq 10^5$
- $1 \leq C_i \leq 10^6$
- $1 \leq i \leq j \leq N$

### Saída

A saída consiste em  $Q$  linhas. Em cada linha deve-se imprimir um único inteiro correspondendo a soma total de todas as salas que não estão entre o intervalo  $(i,j)$ .

Exemplo de Entrada	Exemplo de Saída
5 4	10
1 2 3 4 5	14
2 3	0
1 1	3
1 5	
3 5	

Para este caso, temos 5 salas de tesouros onde em cada sala temos  $[1, 2, 3, 4, 5]$  moedas de ouro, respectivamente. Na primeira pergunta o intervalo  $(2,3)$  está inacessível, logo só temos acesso às salas 1, 4 e 5, onde a soma total de moedas é 10. Para a segunda pergunta apenas a sala 1 não está acessível, logo temos  $2 + 3 + 4 + 5 = 14$ . Na terceira pergunta, nenhuma sala está acessível, logo a resposta é 0. Na última pergunta, somente as salas 1 e 2 está acessível, logo a resposta é  $1 + 2 = 3$ .

## 7 Problema G: Gado

Arquivo: G.[c|cpp|java|py]

Tempo limite: 2s

Rael adora programar, e em sua última jornada de estudo no site CodeHorses ele se deparou com um problema que não sabia como resolver!

### Problema

Inicialmente temos uma string  $S$ , e  $Q$  consultas. Uma consulta consiste em contar o número máximo de pares interessantes em uma subsequência de um intervalo da string  $S$ . Um par interessante pode ser de dois tipos: de parênteses ou de colchetes.

Um par de parênteses  $(i,j)$  é dito interessante se o  $i$ -ésimo caractere da string  $S$  é uma abertura de parênteses ‘(’ e o  $j$ -ésimo caractere da string  $S$  é um fechamento de parênteses ‘)’, onde  $i < j$ .

Analogamente, um par de colchetes  $(i,j)$  é dito interessante se o  $i$ -ésimo caractere da string  $S$  é uma abertura de colchetes ‘[’ e o  $j$ -ésimo caractere da string  $S$  é um fechamento de colchetes ‘]’, onde  $i < j$ .

Note que cada índice só pode aparecer em um único par.

Como Rael está muito ocupado estudando para entrar no PUBG (Partido da União dos Barões do Gado), ele pediu a sua ajuda para resolver esse problema!

### Entrada

A primeira linha da entrada possui o inteiro  $N$ , referente ao tamanho da string  $S$ . A segunda linha da entrada possui a string  $S$  de tamanho  $N$ . A terceira linha da entrada possui o inteiro  $Q$ , referente ao número de consultas para serem respondidas. Seguem-se  $Q$  linhas, cada uma delas contendo 2 valores  $i$  e  $j$ , referentes ao intervalo da pergunta.

### Restrições

- $1 \leq N \leq 10^5$
- $1 \leq Q \leq 10^5$
- $1 \leq i \leq j \leq N$

### Saída

A saída consiste em  $Q$  linhas. Em cada linha deve-se imprimir um único inteiro correspondendo ao número máximo de pares interessantes de parênteses seguido do número máximo de pares interessantes de colchetes entre o intervalo  $(i,j)$ .



Exemplo de Entrada	Exemplo de Saída
6 (([])) 3 1 6 3 6 1 4	2 1 0 1 1 0

Na primeira consulta temos o intervalo que compreende a string  $S$  inteira. Podemos formar os seguintes pares de parênteses interessantes:  $(1, 6)$  e  $(2, 4)$ . Para os pares de colchetes interessantes temos:  $(3, 5)$ .

No segundo caso, o intervalo compreende a seguinte substring de  $S$ :  $[])$ . Pode-se observar que nesse caso não é possível formar nenhum par de parênteses interessante, apenas um único par de colchetes interessante dado pelos índices  $(3, 5)$ .

## 8 Problema H: Kangaroo Word

Arquivo: H.[c|cpp|java|py]

Tempo limite: 5s

Felipera gosta muito de redes sociais e de confraternizar com os seus amigos pela internet! Em um de seus encontros virtuais, ele descobriu um novo conceito e decidiu compartilhar com seus amigos virtuais. Ele descobriu que existem certas palavras que possuem sinônimos que são subsequências da própria palavra, por exemplo: **LONE** é canguru de **ALONE**, e **MALE** é canguru de **MASCULINE**. Eles gostaram tanto do conceito que decidiram fazer um problema para testar os conhecimentos de Felipera.

### Problema

Inicialmente, são dadas  $N$  palavras que servem como o dicionário do problema. Em seguida são definidas  $M$  relações de sinônimos entre as palavras do dicionário. Seguem-se então  $Q$  consultas, em cada consulta é fornecida uma palavra  $S$  (pertencente ao dicionário) e deseja-se saber se existe uma palavra distinta de  $S$  que é quase-canguru de  $S$  que esteja no dicionário. Uma palavra  $S$  é quase-canguru de  $T$  se ela respeita as seguintes propriedades:

1.  $S$  é um prefixo de  $T$
2. O tamanho da palavra  $S$  é menor que o tamanho da palavra  $T$
3.  $S$  e  $T$  são sinônimos

Como Felipera está ocupado jogando jogos de ritmo, ele decidiu pedir a sua ajuda!

### Entrada

A primeira linha contém um inteiro  $N$ , referente ao número de palavras no dicionário. Nas próximas  $N$  linhas, seguem-se as  $N$  palavras do dicionário. A próxima linha contém um inteiro  $M$ , referente ao número de relações entre as palavras do dicionário. Nas próximas  $M$  linhas, seguem-se 2 inteiros  $U$  e  $V$ , indicando que a palavra de índice  $U$  e a palavra de índice  $V$  são sinônimos entre si. A próxima linha contém um inteiro  $Q$ , referente ao número de consultas a serem feitas. As próximas  $Q$  linhas contém um inteiro  $X$ , referente ao índice da palavra  $X$ .

### Restrições

- $1 \leq N \leq 10^5$
- $1 \leq M \leq \min(N * (N - 1)/2, 10^5)$
- $1 \leq Q \leq 10^5$
- $1 \leq SUM \leq 10^6$
- $1 \leq X, U, V \leq N$

SUM é o somatório do tamanho de todas as palavras do dicionário.  
Note que todas as palavras do dicionário são distintas.

## Saída

Para cada inteiro  $X$  nas  $Q$  consultas, deve-se responder o menor índice de alguma palavra que esteja no dicionário e seja quase-canguru a palavra de índice  $X$ .

Caso não existe nenhuma palavra quase-canguru a palavra de índice  $X$ , imprima -1.

Exemplo de Entrada	Exemplo de Saída
5	5
banana	-1
chicken	2
chickenidos	-1
clicker	-1
bana	
3	
1 2	
2 3	
2 5	
5	
1	
2	
3	
4	
5	

Para esta entrada, temos que banana é sinônimo de chicken que é sinônimo de chickenidos e de bana.

Para a primeira consulta, “bana” é uma palavra quase-canguru de “banana”, já que “bana” é um prefixo de “banana” e como “banana” é sinônimo de “chicken” e “chicken” é sinônimo de “bana”, temos que “banana” é sinônimo de “bana” e por isso “bana” é quase-canguru de “banana”. Como o índice de “bana” é 5, a resposta é 5.

Para a segunda consulta, não existe nenhuma palavra quase-canguru de chicken, por isso a resposta é -1.

No terceiro caso, temos que “chicken” é uma palavra quase-canguru de “chickenidos” e, portanto, a resposta é 2.

## 9 Problema I: Cracker

Arquivo: I.[c|cpp|java|py]

Tempo limite: 2s

Adam adora navegar pela internet e explorar o conteúdo de diversos sites educativos e totalmente legalizados. Em um dia, enquanto estava estudando e fazendo boas ações online, Adam “acidentalmente” começou a navegar em sites duvidosos, e foi infectado pelo vírus **EXPO\***. O vírus **EXPO** se comporta da seguinte maneira: Para cada computador infectado, no final do dia este computador infecta um outro que ainda não foi infectado, dessa maneira, começamos com 1 computador infectado no primeiro dia, no segundo dia temos 2 computadores infectados, no terceiro dia temos 4 computadores infectados, e assim por diante.

### Problema

Adam, percebendo o que havia feito, decide estimar o impacto que o vírus pode causar depois de  $N$  dias. Contudo, Adam faltava as aulas durante a graduação e não sabe como estimar, por isso pediu sua ajuda!

### Entrada

A entrada possui um inteiro  $N$  referente ao número de dias que se passaram desde que o computador de Adam foi contaminado.

### Restrições

- $1 \leq N \leq 30$

### Saída

A saída deve possuir um inteiro, referente ao número de computadores que foram infectados depois dos  $N$  dias.

Exemplo de Entrada	Exemplo de Saída
1	2
2	4
3	8

\* Emissário de Xerxes, o Poderoso Ovão.

## 10 Problema J: Tamanho da Janela

Arquivo: J.[c|cpp|java|py]

Tempo limite: 2s

Em momentos livre, prof. ClaudiusLinus vai as madeireiras comprar restos de madeiras que lhe possam ser útil para seus afazeres domésticos, cons fins de reciclagem de material. Os restos de madeiras são colocados num canto da madeireira. Cada pedaço de madeira tem um valor que pode ser positivo ou negativo (só servem para lenha, as vezes nem isto). Assim os restos estão numa sequência de  $M$  valores. Ao examinar estes restos, prof. ClaudiusLinus que saber qual a quantidade de madeira nesta sequência ele pode comprar, e seu valor máximo desta sub-sequência.

### Problema

*Parente* de um dos problemas do *warmup*, este é encontrar o tamanho de uma janela ( $W$ ) que encontre a soma máxima de uma sequência.

### Entrada

A primeira linha de entrada contém um único inteiro positivo,  $N$  indicando o número de casos de testes. Cada um dos  $N$  casos de testes seguem o seguinte formato: uma linha com um inteiro positivo  $M$  que representa o número de pedaços de madeira. Na linha seguinte, segue os  $M$  inteiros representando a qualidade de cada pedaço de madeira.

### Restrições

- $1 < N \leq 100$
- $1 \leq M \leq 100$
- $-10000 \leq X \leq 10000$

### Saída

Para cada caso de teste, encontre e apresente o tamanho desta janela  $W$  que forneça o ganho máximo da sequência  $M$ . Para formatação, siga os exemplos abaixo, com um “\n” ao final de cada linha na saída.

Exemplo de Entrada	Exemplo de Saída
4	Caso:1    Janela:2    SOMA: 13
4	Caso:2    Janela:2    SOMA: 15
5 -16 6 7	Caso:3    Janela:2    SOMA: 211
2	Caso:4    Janela:1    SOMA: 9
7 8	
10	
-999 -28 -11 32 -23 9 10 -111 12 199	
2	
9 -9	

## 11 Problema K: Dois malucos

Arquivo: K.[c|cpp|java|py]

Tempo limite: 2s

Peter e Vinicius adoram visitar a sua avó Kaqui. Como ela adora seus netos, sempre os presenteia com uns trocados devido ao seus grandes esforços em ajudá-la! Na sua próxima visita, os meninos estão planejando levar um presente a sua avó. O presente consiste em uma cesta com uma quantidade enorme de café, pois eles souberam que o querido café de sua avó havia acabado. Peter e Vinicius, como adoram muito sua avó, acabaram levando mais café do que deveriam. Assim, devido ao peso da cesta, cada um dos meninos não consegue dar  $K$  ou mais passos consecutivos enquanto carrega a cesta. Por mais que tenham contribuído de maneira igual para o presente, tanto Peter quanto Vinicius querem entregar a cesta a sua avó. Assim, eles criaram o seguinte jogo para determinar quem entregará a cesta.

### Problema

O jogador atual (aquele que está com a cesta), escolhe um número de passos menor que  $K$  e então andará essa quantia (maior que zero) em direção a casa de sua avó, e então irá passar a cesta para o outro menino, repetindo-se o processo até chegarem à casa da avó. Aquele que chegar na casa da avó com a cesta em mãos poderá entregá-la para ela. Atualmente, Peter está com a cesta e os meninos estão à  $N$  passos de distância de sua avó e se perguntaram, se ambos jogarem de forma ótima, quem entregará a cesta a avó?

### Entrada

A entrada é dada por uma linha contendo 2 inteiros  $N$  e  $K$ , onde  $N$  é a distância (em passos) da casa de sua avó, e  $K$  é o número de passos que excede sua capacidade física dos meninos.

### Restrições

- $1 \leq N \leq 10^5$
- $1 \leq K \leq 10^3$

### Saída

Imprima “Peter” (sem aspas) caso o vencedor seja o Peter. Caso contrário imprima “Vinicius” (sem aspas).

Exemplo de Entrada	Exemplo de Saída
3 3	Vinicius
4 3	Peter
5 3	Peter

Explicação da primeira entrada: → Peter começa dando 1 passo, Vinicius recebe a cesta à 2 passos de distância da casa de sua avó, assim, Vinicius dá 2 passos e chega a casa da avó.  
→ Peter começa dando 2 passos, Vinicius recebe a cesta à 1 passo de distância da casa de sua avó, assim, Vinicius dá 1 passo e chega a casa da avó.