

A. Múltiplos

Time limit: 1s

Leia 2 valores inteiros (A e B). Após, o programa deve mostrar uma mensagem **"Sao Multiplos"** ou **"Nao sao Multiplos"**, indicando se os valores lidos são múltiplos entre si.

Entrada

A entrada contém valores inteiros.

Saída

A saída deve conter uma das mensagens conforme descrito acima.

Exemplo de Entrada	Exemplo de Saída
6 24	Sao Multiplos
6 25	Nao sao Multiplos

Adaptado por Neilor Tonin, URI  Brasil

B. The Overdosing Ubiquity

Time limit: 1s

Memory limit: 256 MB

The fundamental prerequisite for justice is not to be correct, but to be strong. That's why justice is always the victor.

The Cinderswarm Bee. Koyomi knows it.

The bees, according to their nature, live in a tree. To be more specific, a *complete binary tree* with n nodes numbered from 1 to n . The node numbered 1 is the root, and the parent of the i -th ($2 \leq i \leq n$) node is $\lfloor \frac{i}{2} \rfloor$. Note that, however, all edges in the tree are undirected.

Koyomi adds m extra undirected edges to the tree, creating more complication to trick the bees. And you're here to count the number of *simple paths* in the resulting graph, modulo $10^9 + 7$. A *simple path* is an alternating sequence of adjacent nodes and undirected edges, which begins and ends with nodes and does not contain any node more than once. Do note that a single node is also considered a valid *simple path* under this definition. Please refer to the examples and notes below for instances.

Input

The first line of input contains two space-separated integers n and m ($1 \leq n \leq 10^9$, $0 \leq m \leq 4$) — the number of nodes in the tree and the number of extra edges respectively.

The following m lines each contains two space-separated integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — describing an undirected extra edge whose endpoints are u and v .

Note that there may be multiple edges between nodes in the resulting graph.

Output

Output one integer — the number of *simple paths* in the resulting graph, modulo $10^9 + 7$.

Examples

input
3 0
output
9

input
3 1
2 3
output
15

input
2 4
1 2
2 1
1 2
2 1

output
12

Note

In the first example, the paths are: (1); (2); (3); (1, 2); (2, 1); (1, 3); (3, 1); (2, 1, 3); (3, 1, 2). (For the sake of clarity, the edges between nodes are omitted since there are no multiple edges in this case.)

In the second example, the paths are: (1); (1, 2); (1, 2, 3); (1, 3); (1, 3, 2); and similarly for paths starting with 2 and 3. ($5 \times 3 = 15$ paths in total.)

In the third example, the paths are: (1); (2); any undirected edge connecting the two nodes travelled in either direction. ($2 + 5 \times 2 = 12$ paths in total.)

C. O Maior

Time limit: 1s

Faça um programa que leia três valores e apresente o maior dos três valores lidos seguido da mensagem “eh o maior”. Utilize a fórmula:

$$MaiorAB = \frac{(a+b+abs(a-b))}{2}$$

Entrada

O arquivo de entrada contém três valores inteiros.

Saída

Imprima o maior dos três valores seguido por um espaço e a mensagem "eh o maior".

Exemplos de Entrada	Exemplos de Saída
7 14 106	106 eh o maior
217 14 6	217 eh o maior

Adaptado por Neilor Tonin, URI  Brasil

D. Rede ótica

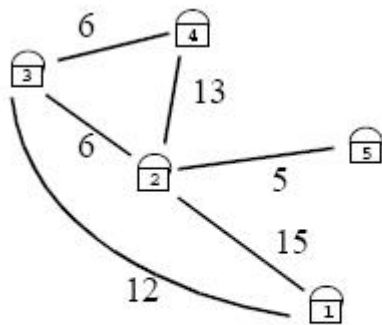
Time limit: 0.157s

Memory limit: 1536 MB

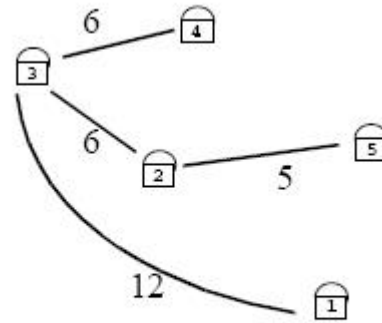
Os caciques da região de Tutuaçu pretendem integrar suas tribos à chamada “aldeia global”. A primeira providência foi a distribuição de telefones celulares a todos os pajés. Agora, planejam montar uma rede de fibra ótica interligando todas as tabas. Esta empreitada requer que sejam abertas novas picadas 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.



Ramos de rede possíveis com custo ambiental associado



Interligação das tabas com menor custo ambiental

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 contêm 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 $N = 0$.

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 $X < Y$. 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

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:

Teste 1

```
1 2
1 3
```

Teste 2

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

Restrições

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

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

$1 \leq X \leq 100$

$1 \leq Y \leq 100$

$1 \leq Z \leq 100$

Olimpiada Brasileira de Informatica 2000

E. Sorvete

Time limit: 0.116s

Memory limit: 1536 MB

Joãozinho é um menino que costuma ir à praia todos os finais de semana com seus pais. Eles freqüentam sempre a mesma praia, mas cada semana o pai de Joãozinho estaciona o carro em um local diferente ao longo da praia, e instala sua família em um ponto na praia em frente ao carro. Joãozinho é muito comilão, e adora de tomar sorvete na praia. Contudo, alguns dias acontece de nenhum sorveteiro passar pelo local onde eles estão. Intrigado com isto, e não querendo mais ficar sem tomar seu sorvete semanal, Joãozinho foi até a Associação dos Sorveteiros da Praia (ASP), onde ficou sabendo que cada sorveteiro passa o dia percorrendo uma mesma região da praia, indo e voltando. Além disto, cada sorveteiro percorre todos os dias a mesma região. Joãozinho conseguiu ainda a informação dos pontos de início e fim da região percorrida por cada um dos sorveteiros.

Com base nestes dados, Joãozinho quer descobrir os locais da praia onde o pai dele deve parar o carro, de forma que pelo menos um sorveteiro passe naquele local. Só que o volume de dados é muito grande, e Joãozinho está pensando se seria possível utilizar o computador para ajudá-lo nesta tarefa. No entanto Joãozinho não sabe programar, e está pedindo a sua ajuda.

Tarefa

Você deve escrever um programa que leia os dados obtidos pelo Joãozinho e imprima uma lista de intervalos da praia por onde passa pelo menos um sorveteiro.

Entrada

Seu programa deve ler vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois inteiros não negativos, P e S , que indicam respectivamente o comprimento em metros da praia e o número de sorveteiros. Seguem-se S linhas, cada uma contendo dois números inteiros U e V que descrevem o intervalo de trabalho de cada um dos sorveteiros, em metros contados a partir do início da praia ($U < V$, $0 \leq U \leq P$ e $0 \leq V \leq P$). O final da entrada é indicado por $S = 0$ e $P = 0$.

Exemplo de Entrada

```
200 2
0 21
110 180
1000 3
10 400
80 200
400 1000
10 2
1 4
5 6
0 0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir uma lista dos intervalos da praia que são servidos por pelo menos um sorveteiro. 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. Cada intervalo da lista deve aparecer em uma linha separada, sendo descrito por dois números inteiros

U e V , representando respectivamente o início e o final do intervalo ($U < V$) . O final da lista de intervalos deve ser indicado por uma linha em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

Teste 1

0 21

110 180

Teste 2

10 1000

Teste 3

1 4

5 6

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq P \leq 10000$ (P = 0 apenas para indicar o final da entrada)

$0 \leq S \leq 5000$ (S = 0 apenas para indicar o final da entrada)

$0 \leq U < V \leq P$

Olimpiada Brasileira de Informatica 2001

F. Green and Black Tea

Time limit: 1s

Memory limit: 256 MB

Innokentiy likes tea very much and today he wants to drink exactly n cups of tea. He would be happy to drink more but he had exactly n tea bags, a of them are green and b are black.

Innokentiy doesn't like to drink the same tea (green or black) more than k times in a row. Your task is to determine the order of brewing tea bags so that Innokentiy will be able to drink n cups of tea, without drinking the same tea more than k times in a row, or to inform that it is impossible. Each tea bag has to be used exactly once.

Input

The first line contains four integers n , k , a and b ($1 \leq k \leq n \leq 10^5$, $0 \leq a, b \leq n$) — the number of cups of tea Innokentiy wants to drink, the maximum number of cups of same tea he can drink in a row, the number of tea bags of green and black tea. It is guaranteed that $a + b = n$.

Output

If it is impossible to drink n cups of tea, print "NO" (without quotes).

Otherwise, print the string of the length n , which consists of characters 'G' and 'B'. If some character equals 'G', then the corresponding cup of tea should be green. If some character equals 'B', then the corresponding cup of tea should be black.

If there are multiple answers, print any of them.

Examples**input**

5 1 3 2

output

GBGBG

input

7 2 2 5

output

BBGBGBB

input

4 3 4 0

output

NO

G. Soma Simples

Time limit: 1s

Leia dois valores inteiros, no caso para variáveis A e B. A seguir, calcule a soma entre elas e atribua à variável **SOMA**. A seguir escrever o valor desta variável.

Entrada

O arquivo de entrada contém 2 valores inteiros.

Saída

Imprima a variável **SOMA** com todas as letras maiúsculas, com um espaço em branco antes e depois da igualdade seguido pelo valor correspondente à soma de A e B. Como todos os problemas, não esqueça de imprimir o fim de linha após o resultado, caso contrário, você receberá "Presentation Error".

Exemplos de Entrada	Exemplos de Saída
30 10	SOMA = 40
-30 10	SOMA = -20
0 0	SOMA = 0

Adaptado por Neilor Tonin, URI  Brasil

H. Comércio de Vinhos na Gergóvia

Time limit: 2s

Como você deve saber do cômico "Astérix e o Escudo de Arverne", Gergóvia consiste de uma única rua e cada habitante da cidade é um vendedor de vinho. Você quer saber como essa economia funciona? Bem simples: todos compram vinhos dos outros habitantes da cidade. Cada dia, cada habitante decide quanto vinho ele quer comprar ou vender. Curiosamente, a demanda e o fornecimento são sempre os mesmos de modo que cada habitante consegue o que deseja.

Contudo, há um problema: transportar o vinho de uma casa para outra resulta em trabalho. Sendo todos os vinhos igualmente bons, os habitantes de Gergóvia não se importam com quais pessoas eles irão comercializar, eles estão somente interessados em vender e comprar um quantidade específica de vinho. Eles são espertos o suficiente para imaginar uma forma de negociar de modo que todo o montante de trabalho necessário para o transporte seja minimizado.

Nesse problema você está sendo inquerido para reconstruir o comércio durante um dia em Gergóvia. Para simplificar, nós assumimos que as casas são construídas ao longo de uma linha reta com a mesma distância entre as casas adjacentes. Transportar uma garrafa de vinho de uma casa para uma casa adjacente resulta em uma unidade de trabalho.

Entrada

A entrada consiste de vários casos de teste.

Cada caso de teste inicia com o número de habitantes n ($2 \leq n \leq 100000$). A linha seguinte contém n inteiros a_i ($-1000 \leq a_i \leq 1000$). Se $a_i \geq 0$, isso significa que cada habitante que vive na i th casa, deseja comprar a_i garrafas de vinho, caso contrário se $a_i < 0$, ele deseja vender $-a_i$ garrafas de vinho. Você pode assumir que os números a_i resumem a 0.

O último caso de teste é seguido por uma linha contendo 0.

Saída

Para cada caso de teste, imprima a quantidade mínima de unidades de trabalho necessárias para que todo habitante tenha sua demanda cumprida. Você pode assumir que este número cabe em um inteiro de 64 bits com sinal (em C/C++ você pode usar o tipo de dados "long long", em JAVA o tipo de dados "long").

Exemplo de Entrada	Exemplo de Saída
5 5 -4 1 -3 1 6 -1000 -1000 -1000 1000 1000 1000 0	9 9000

Univeristy of Ulm Local Contest 2006/2007

Contest Local, Universidade de Ulm  Alemanha