

## Mục lục

$a + b = c$ — ADD . . . . .	1
Largest weight subsequence — SUBSEQMAX . . . . .	2
Sorting a sequence — SORT . . . . .	3
Sorting a sequence by the combsort algorithm — COMBSORT . . . . .	4
Checking parentheses expressions — PARENTHESES . . . . .	5
Water jug problem — WATERJUG . . . . .	6
Holey N-Queens — HOLEYQUEENS . . . . .	7
Easy modified sudoku — EZSUDOKU . . . . .	8
Bloper Operator — BLOPER . . . . .	10
Wine trading in Gergovia — GERGOVIA . . . . .	11
I am very busy — BUSYMAN . . . . .	12
Aggressive cows — AGGRCOW . . . . .	14
Eko — EKO . . . . .	15
Copying Books — BOOKS1 . . . . .	16
Tìm xâu con chung dài nhất — LCS . . . . .	17
Khoảng cách hai xâu — ED . . . . .	18
Tìm tập độc lập cực đại trên cây — TMAXSET . . . . .	19
Searching in graphs - GSEARCH . . . . .	20
Minimum spanning tree - MST . . . . .	21
The shortest path - MINPATH . . . . .	22

## Problem A. $a + b = c$

Input File Name: **stdin**  
Output File Name: **sdout**  
Time Limit: 1 s  
Memory Limit: 256 MB

### Input

Consist in one line 2 positive integer numbers  $a$  and  $b$  ( $a, b < 10^{19}$ )

### Output

Write in one line the resulting number  $c$ .

### Examples

stdin	sdout
1 2	3

## Problem B. Largest weight subsequence

Input File Name: `stdin`  
Output File Name: `sdout`  
Time Limit: 1 s  
Memory Limit: 256 MB

Find the largest weight subsequence of a given sequence of numbers

- Given a sequence  $s = \langle a_1, \dots, a_n \rangle$
- a subsequence is  $s(i, j) = \langle a_i, \dots, a_j \rangle$ ,  $1 \leq i \leq j \leq n$
- weight  $w(s(i, j)) =$

$$\sum_{k=i}^j a_k$$

- Problem: find the subsequence having largest weight

### Input

- The first line contains one integer number  $n \leq 10^6$ .
- The second line contains  $n$  integer numbers.

### Output

Write uniquely the weight of the found sequence.

### Examples

stdin	sdout
6 -2 11 -4 13 -5 2	20

## Problem C. Sorting a sequence

Input File Name: `stdin`  
Output File Name: `sdout`  
Time Limit: 1 s  
Memory Limit: 256 MB

### Input

- The first line contains one integer number  $n \leq 10^6$ .
- The second line contains  $n$  real numbers, each one has exactly 2 digits after the floating point.

### Output

Write in one line the increasing sorted sequence, each one has exactly 2 digits after the floating point.

### Examples

stdin	sdout
6 2.22 5.25 6.26 1.21 4.24 3.23	1.21 2.22 3.23 4.24 5.25 6.26

## Problem D. Sorting a sequence by the combsort algorithm

Input File Name: `stdin`  
Output File Name: `sdout`  
Time Limit: 1 s  
Memory Limit: 256 MB

### Input

- The first line contains one integer number  $n \leq 10^6$ .
- The second line contains  $n$  real numbers, each one has exactly 2 digits after the floating point.

### Output

Write in one line the increasing sorted sequence, each one has exactly 2 digits after the floating point.

### Examples

stdin	sdout
6 2.22 5.25 6.26 1.21 4.24 3.23	1.21 2.22 3.23 4.24 5.25 6.26

## Problem E. Checking parentheses

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 1 s  
Memory Limit: 256 MB

### Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer  $T$  and is not greater than 1000. Each of  $T$  following lines describes a parentheses expression including: `'(',')'`, `'[],[]'`, `'{','}'`.

### Output

For each dataset, write in one line 1 or 0 if the expression is correct or not respectively.

### Examples

stdin	stdout
2 ( ) [] [ ( )	1 0

### Scoring

The length of each expression  $\leq 10^5$ .

## Problem F. Water jug problem

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 1 s  
Memory Limit: 256 MB

There are two jugs, a  $a$ -gallon one and a  $b$ -gallon one ( $a, b$  are positive integer). There is a pump with unlimited water. Neither jug has any measuring marking on it. How can you get exactly  $c$ -gallon jug ( $c$  is a positive integer)?

### Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer  $T$  and is not greater than 1000. Each of  $T$  following lines consists 3 positive integer numbers  $a, b, c \leq 10^8$ .

### Output

For each dataset, write in one line one integer which is the minimum number of water movement steps to get  $c$  gallon. Write -1 if there is no way to get  $c$ .

### Examples

stdin	stdout
2	2
3 8 5	-1
3 4 5	

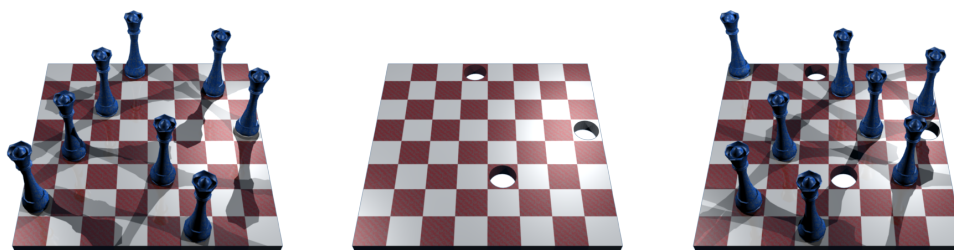
### Scoring

There are half of total test cases having  $a, b, c \leq 10^3$ .

## Problem G. Holey N-Queens

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 1 s  
Memory Limit: 1024 MB

The  $N$ -queens problem is the problem of placing  $N$  queens on a  $N \times N$  chessboard so that no queen shares a row, column or a diagonal with any other queen. Essentially, we are trying to place the queens without any queen threatening another. For example, the first image below (without holes in the board) is a solution to the 8-queens problem.



For this problem, consider the problem we'll call the 'holey  $N$ -queens problem'. Instead of having an everyday chessboard (of arbitrary size), your chessboard is like the second image above (without queens): it has holes through some of the squares. You can't place a queen on a square that has a hole, but a queen can threaten another even if there is a hole on the path between them. Given a holey chessboard, you must find placements for the  $N$  queens so that no queen threatens another. The third image above (with holes and queens) shows one such solution.

### Input

Input consists of up to 1000 board descriptions. Each description starts with a line containing a pair of integers,  $3 \leq N \leq 12$  and  $0 \leq M \leq N^2$ , indicating respectively the size of one side of the board, and the number of holes. The next  $M$  lines each describe the location of a unique hole in the board. A hole is described by its row and column, each in the range  $[1, N]$ . The end of input is marked by values of zero for  $N$  and  $M$ .

### Output

For each problem description, print out the number of unique  $N$ -queens solutions that are possible. Two solutions are considered different if any space is occupied by a queen in one solution but not in the other.

### Examples

stdin	stdout
8 0	92
8 3	50
1 4	
6 5	
4 8	
0 0	

## Problem H. Easy modified sudoku

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 1 s  
Memory Limit: 256 MB

Do you know the old famous game, sudoku? It is a logic-based,combinatorial number-placement puzzle. The objective is to fill a  $9 \times 9$  grid with digits so that each column, each row, and each of the nine  $3 \times 3$  sub-grids that compose the grid (also called "boxes", "blocks", "regions", or "sub-squares") contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a unique solution.

Completed puzzles are always a type of Latin square with an additional constraint on the contents of individual regions. For example, the same single integer may not appear twice in the same row, column or in any of the nine  $3 \times 3$  subregions of the  $9 \times 9$  playing board. French newspapers featured variations of the puzzles in the 19th century, and the puzzle has appeared since 1979 in puzzle books under the name Number Place.However, the modern Sudoku only started to become mainstream in 1986 by the Japanese puzzle company Nikoli, under the name Sudoku, meaning single number. It first appeared in a US newspaper and then The Times (UK) in 2004, from the efforts of Wayne Gould, who devised a computer program to rapidly produce distinct puzzles.

Since the Sudoku game is pretty hard, Mr. Bruce Banner, who is also fond in sudoku games, creates a new modified version of normal sudoku. It is tiled by  $8 \times 8$  grids and divided into 4 areas of  $4 \times 4$  grids. The rule is also simplified. You need to place numbers from 1 to 8 to the empty grids(denoted by a number 0) to the sudoku puzzle. Each number in the same columns or row must be distinct. In each area of  $4 \times 4$  grids, it is allowed to have duplicate numbers up to 2 numbers.

Now, your task is to help Mr. Bruce Banner to determine whether the modified sudoku puzzle he has created is of a valid puzzle or not. If it is a valid puzzle, you will also need to output the solution.

Note: It is guaranteed that here is a unique answer in each test cases.

### Input

The first line of input contains an integer  $T(1 \leq T \leq 100)$ , the number of testcases. Then, it will be followed by  $8 * T$  lines which contain the modified sudoku puzzles. Each testcases will be separated by a newline.

### Output

For each case print "Test case #X: ", where  $X(1 \leq X \leq T)$  is the case number, followed by an answer "YES" or "NO" and a newline. If there exists a valid solution, print the solution of each puzzles, and then print a newline.



## Examples

stdin	stdout
2	Test case #1: YES
1 2 3 4 5 6 7 0	1 2 3 4 5 6 7 8
5 6 2 1 3 4 0 7	5 6 2 1 3 4 8 7
6 5 0 7 2 1 3 4	6 5 8 7 2 1 3 4
3 4 7 8 6 5 2 1	3 4 7 8 6 5 2 1
2 1 5 6 8 7 4 3	2 1 5 6 8 7 4 3
4 3 6 5 7 8 1 2	4 3 6 5 7 8 1 2
0 7 1 2 4 3 5 6	8 7 1 2 4 3 5 6
7 0 4 3 1 2 6 5	7 8 4 3 1 2 6 5
1 1 1 1 1 1 1 1	Test case #2: NO
1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1	
1 1 1 1 1 1 1 1	

## Problem I. Bloper Operators

Input File Name:       Standard Input  
Output File Name:      Standard Output  
Time Limit:            1s  
Memory Limit:          256 MB

Given a set of  $N$  integer  $A = \{1, 2, 3, \dots, N\}$  and an integer  $S$ , your task is find a way to insert an operator '+' or '-' to every neighbor pair of  $A$  or before 1, that the result of the expression after insert equal to  $S$ .

### Input

The input file consists of several datasets. The first line of the input file contains the number  $T$  of datasets which is a positive integer and is not greater than 20. Each one of the following lines describe a dataset: Write in a single line,  $N$  and  $S$  ( $1 \leq N \leq 500, |S| \leq 125250$ ).

### Output

For each line in  $T$  lines, write 1 if there are way(s) to insert, otherwise write 0.

### Examples

Standard Input	Standard Output
2	1
9 5	0
5 6	

### Explanation

One way to insert operators for the first testcase:  $1-2+3-4+5-6+7-8+9=5$

## Problem J. Wine trading in Gergovia

Input File Name:       Standard Input  
Output File Name:       Standard Output  
Time Limit:            1s  
Memory Limit:          256 MB

Gergovia consists of one street, and every inhabitant of the city is a wine salesman. Everyone buys wine from other inhabitants of the city. Every day each inhabitant decides how much wine he wants to buy or sell. Interestingly, demand and supply is always the same, so that each inhabitant gets what he wants.

There is one problem, however: Transporting wine from one house to another results in work. Since all wines are equally good, the inhabitants of Gergovia don't care which persons they are doing trade with, they are only interested in selling or buying a specific amount of wine.

In this problem you are asked to reconstruct the trading during one day in Gergovia. For simplicity we will assume that the houses are built along a straight line with equal distance between adjacent houses. Transporting one bottle of wine from one house to an adjacent house results in one unit of work.

### Input

The input consists of several test cases.

Each test case starts with the number of inhabitants  $N$  ( $2 \leq N \leq 100000$ ).

The following line contains  $n$  integers  $a_i$  ( $-1000 \leq a_i \leq 1000$ ).

If  $a_i \geq 0$ , it means that the inhabitant living in the  $i$ th house wants to buy  $a_i$  bottles of wine. If  $a_i < 0$ , he wants to sell  $-a_i$  bottles of wine.

You may assume that the numbers  $a_i$  sum up to 0.

The last test case is followed by a line containing 0.

### Output

For each test case print the minimum amount of work units needed so that every inhabitant has his demand fulfilled.

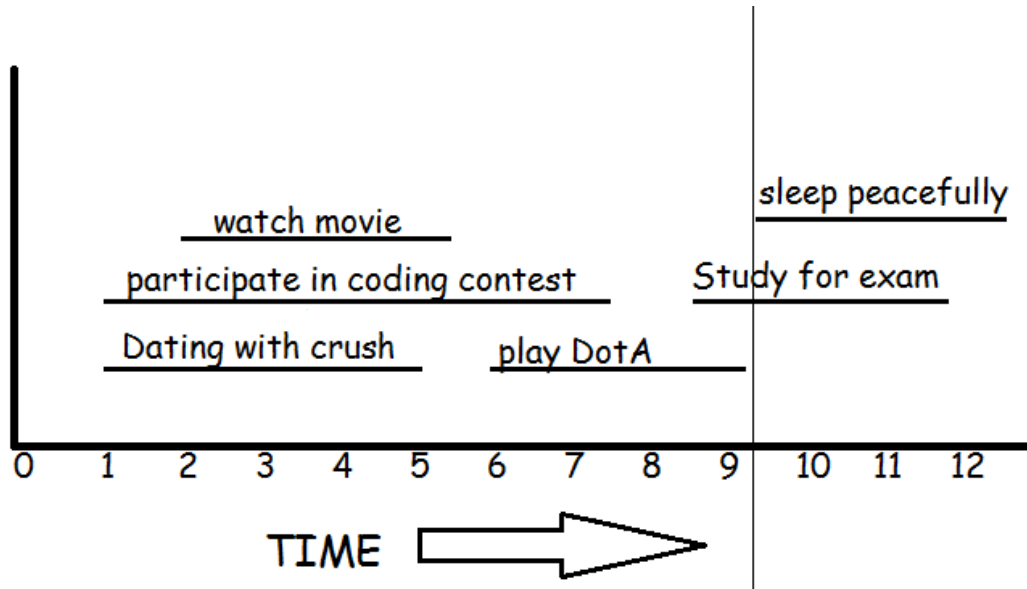
### Examples

Standard Input	Standard Output
5	9
5 -4 1 -3 1	9000
6	
-1000 -1000 -1000 1000 1000 1000	
0	

## Problem K. I am very busy

Input File Name: Standard Input  
Output File Name: Standard Output  
Time Limit: 1s  
Memory Limit: 256 MB

You are actually very busy man. You have a big schedule of activities. Your aim is to do as much as activities as possible.



In the given figure, if you go to date with crush, you cannot participate in the coding contest and you can't watch the movie. Also if you play DotA, you can't study for the exam. If you study for the exam you can't sleep peacefully. The maximum number of activities that you can do for this schedule is 3.

Either you can

- watch movie, play DotA and sleep peacefully (or)
- date with crush, play DotA and sleep peacefully

### Input

The first line consists of an integer  $T$ , the number of test cases. For each test case the first line consists of an integer  $N$ , the number of activities. Then the next  $N$  lines contains two integers  $m$  and  $n$ , the start and end time of each activity.

### Output

For each test case find the maximum number of activities that you can do.

### Scoring

$$1 \leq T \leq 10$$

$$1 \leq N \leq 100000$$

$$0 \leq \text{start} < \text{end} \leq 1000000$$

## Examples

Standard Input	Standard Output
3	1
3	2
3 9	3
2 8	
6 9	
4	
1 7	
5 8	
7 8	
1 8	
6	
7 9	
0 10	
4 5	
8 9	
4 10	
5 7	

## Problem L. Aggressive cows

Input File Name:       Standard Input  
Output File Name:      Standard Output  
Time Limit:            1s  
Memory Limit:          256 MB

Farmer John has built a new long barn, with  $N$  ( $2 \leq N \leq 100,000$ ) stalls. The stalls are located along a straight line at positions  $x_1, \dots, x_N$  ( $0 \leq x_i \leq 1,000,000,000$ ).

His  $C$  ( $2 \leq C \leq N$ ) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ wants to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

### Input

$t$  – the number of test cases, then  $t$  test cases follows.

- Line 1: Two space-separated integers:  $N$  and  $C$
- Lines 2  $\dots$   $N + 1$ : Line  $i + 1$  contains an integer stall location,  $x_i$

### Output

For each test case output one integer: the largest minimum distance.

### Examples

Standard Input	Standard Output
1 5 3 1 2 8 4 9	3

### Explanation

FJ can put his 3 cows in the stalls at positions 1, 4 and 8, resulting in a minimum distance of 3.

## Problem M. Eko

Input File Name:       Standard Input  
Output File Name:      Standard Output  
Time Limit:            1s  
Memory Limit:          256 MB

Lumberjack Mirko needs to chop down  $M$  metres of wood. It is an easy job for him since he has a nifty new woodcutting machine that can take down forests like wildfire. However, Mirko is only allowed to cut a single row of trees.

Mirko's machine works as follows: Mirko sets a height parameter  $H$  (in metres), and the machine raises a giant sawblade to that height and cuts off all tree parts higher than  $H$  (of course, trees not higher than  $H$  meters remain intact). Mirko then takes the parts that were cut off. For example, if the tree row contains trees with heights of 20, 15, 10, and 17 metres, and Mirko raises his sawblade to 15 metres, the remaining tree heights after cutting will be 15, 15, 10, and 15 metres, respectively, while Mirko will take 5 metres off the first tree and 2 metres off the fourth tree (7 metres of wood in total).

Mirko is ecologically minded, so he doesn't want to cut off more wood than necessary. That's why he wants to set his sawblade as high as possible. Help Mirko find the maximum integer height of the sawblade that still allows him to cut off at least  $M$  metres of wood.

### Input

The first line of input contains two space-separated positive integers,  $N$  (the number of trees,  $1 \leq N \leq 1000000$ ) and  $M$  (Mirko's required wood amount,  $1 \leq M \leq 2000000000$ ).

The second line of input contains  $N$  space-separated positive integers less than 1 000 000 000, the heights of each tree (in metres). The sum of all heights will exceed  $M$ , thus Mirko will always be able to obtain the required amount of wood.

### Output

The first and only line of output must contain the required height setting.

### Examples

Standard Input	Standard Output
4 7 20 15 10 17	15
5 20 4 42 40 26 46	36

## Problem N. Copying Books

Input File Name:       Standard Input  
Output File Name:      Standard Output  
Time Limit:            1s  
Memory Limit:          256 MB

Before the invention of book-printing, it was very hard to make a copy of a book. All the contents had to be re-written by hand by so called scribes. The scribe had been given a book and after several months he finished its copy. One of the most famous scribes lived in the 15th century and his name was Xaverius Endricus Remius Ontius Xendrianus (Xerox). Anyway, the work was very annoying and boring. And the only way to speed it up was to hire more scribes.

Once upon a time, there was a theater ensemble that wanted to play famous Antique Tragedies. The scripts of these plays were divided into many books and actors needed more copies of them, of course. So they hired many scribes to make copies of these books. Imagine you have  $m$  books (numbered  $1, 2, \dots, m$ ) that may have different number of pages ( $p_1, p_2, \dots, p_m$ ) and you want to make one copy of each of them. Your task is to divide these books among  $k$  scribes,  $k \leq m$ . Each book can be assigned to a single scribe only, and every scribe must get a continuous sequence of books. That means, there exists an increasing succession of numbers  $0 = b_0 < b_1 < b_2, \dots < b_{k-1} \leq b_k = m$  such that  $i$ -th scribe gets a sequence of books with numbers between  $b_{i-1} + 1$  and  $b_i$ . The time needed to make a copy of all the books is determined by the scribe who was assigned the most work. Therefore, our goal is to minimize the maximum number of pages assigned to a single scribe. Your task is to find the optimal assignment.

### Input

The input consists of  $N$  cases (equal to about 200). The first line of the input contains only positive integer  $N$ . Then follow the cases. Each case consists of exactly two lines. At the first line, there are two integers  $m$  and  $k$ ,  $1 \leq k \leq m \leq 500$ . At the second line, there are integers  $p_1, p_2, \dots, p_m$  separated by spaces. All these values are positive and less than 10000000.

### Output

For each case, print exactly one line. The line must contain the input succession  $p_1, p_2, \dots, p_m$  divided into exactly  $k$  parts such that the maximum sum of a single part should be as small as possible. Use the slash character ('/') to separate the parts. There must be exactly one space character between any two successive numbers and between the number and the slash.

If there is more than one solution, print the one that minimizes the work assigned to the first scribe, then to the second scribe etc. But each scribe must be assigned at least one book.

### Examples

Standard Input	Standard Output
2	100 200 300 400 500 / 600 700 / 800
9 3	900
100 200 300 400 500 600 700 800 900	100 / 100 / 100 / 100 100
5 4	
100 100 100 100 100	



## Problem O. Tìm xâu con chung dài nhất — LCS

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 1 giây  
Memory Limit: 1024 MB

Cho hai xâu, hãy tìm xâu con chung dài nhất của hai xâu đó.

### Input

Dòng đầu tiên chứa duy nhất một số  $T \leq 30$  là số lượng bộ test.

Mỗi test gồm 2 dòng, mỗi dòng chứa một xâu gồm các kí tự trong 'a' đến 'z' viết liền nhau, giới hạn độ dài không quá 5000.

### Output

Ghi ra  $T$  dòng, mỗi dòng là kết quả tương ứng cho một test là độ dài xâu con chung dài nhất tìm được.

### Examples

stdin	stdout
1 wigwwnydtyo kwmmka	1

## Problem P. Khoảng cách hai chuỗi — ED

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 1 giây  
Memory Limit: 1024 MB

Cho hai chuỗi, hãy tìm cách biến đổi chuỗi thứ nhất thành chuỗi thứ hai sao cho sử dụng ít phép biến đổi nhất: chèn, sửa, xóa.

### Input

Dòng đầu tiên chứa duy nhất một số  $T \leq 30$  là số lượng bộ test.

Mỗi test gồm 2 dòng, mỗi dòng chứa một chuỗi gồm các kí tự trong ‘a’ đến ‘z’ viết liền nhau, giới hạn độ dài không quá 1500.

### Output

Ghi ra  $T$  dòng, mỗi dòng là kết quả tương ứng cho một test là số lượng phép biến đổi tìm được.

### Examples

stdin	stdout
1 eda f	3

## Problem Q. Tìm tập độc lập cực đại trên cây — TMAXSET

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 0.5 giây  
Memory Limit: 1024 MB

Cho một cây với trọng số trên mỗi đỉnh và một tập con  $Q$  các đỉnh của cây, hãy tìm tập độc lập có tổng trọng số lớn nhất trong  $Q$ .

### Input

Dòng đầu chứa một số  $T \leq 20$  là số lượng bộ test;

Mỗi test có cấu trúc như sau:

- Dòng đầu chứa 2 số  $n$  và  $m$  là số đỉnh và số cạnh của đồ thị;
- Dòng thứ 2 chứa  $n$  số là trọng số trên mỗi đỉnh;
- Từ dòng 3 đến dòng  $m + 2$ , mỗi dòng chứa 2 số là các cặp đỉnh có cạnh nối giữa chúng, dữ liệu đảm bảo đồ thị đã cho là 1 cây;
- Dòng  $m + 3$  chứa 1 số nguyên  $q$  là số truy vấn, sau đó mỗi dòng trong số  $q$  dòng tiếp chứa 1 dãy số:
  - Số đầu tiên trong dãy  $k$  chỉ số lượng của tập đỉnh truy vấn;
  - $k$  số tiếp theo chỉ số hiệu từng đỉnh trong tập (đánh số từ 0) là những đỉnh xét đến trong  $n$  đỉnh ban đầu

### Output

Mỗi truy vấn in ra 1 kết quả là trọng số tập độc lập lớn nhất trên 1 dòng. Sau mỗi một bộ test thì in ra một dòng trống.

### Examples

stdin	stdout
1 3 2 5 4 10 0 2 2 1 1 2 2 1	10

### Explanation

Trong ví dụ trên, có 1 bộ test với cây gồm 3 đỉnh có số hiệu là 0,1,2 và 2 cạnh (0 2) và (2 1). Có 1 truy vấn chỉ xét trên 2 đỉnh có số hiệu 1 và 2.

### Scoring

- 50% số test  $5 \leq n \leq 30$ ,  $1 \leq q \leq 100$ ;
- 25% số test tiếp theo có  $10 \leq n \leq 200$ ,  $100 \leq q \leq 1000$ ;
- 25% số test còn lại có  $150 \leq n \leq 200$ ,  $q = 1000$ .

## Problem R. Searching in graphs

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 0.1 s  
Memory Limit: 512 MB

The problem requires to check the connectivity of a given graph and to find restricted paths between two given vertices  $i$  and  $j$ .

First requirement: Find the number of connected components.

Second requirement: Find the shortest path from  $i$  to  $j$  that has the minimum number of vertices.

Third requirement: Find the path from  $i$  to  $j$  with the smallest lexicographic order.

### Input

- The first line contains two positive integers  $n \leq 10^5$ ,  $m \leq 10^5$ ;
- Each of the  $m$  following lines contains two positive integers  $u$  and  $v$  corresponding to an arc  $(u, v)$ ;
- The following line contains 2 positive integers  $i$  and  $j$  corresponding to the requirements of finding restricted paths between  $i$  and  $j$ .

### Output

- Convert the given directed graph to the corresponding undirected graph by deleting the orientation in each arc. Write out on the first line an integer that is the number of connected components of the undirected graph.
- The second line contains an integer that is the number of vertices on the resulted path by the second requirement. Print 0 if the path does not exists.
- The third line contains an integer  $\ell$  that is the number of vertices on the result path by the third requirement. Print 0 if the path does not exists.
- The fourth line contains  $\ell$  integers that is the labels of the resulted vertices by the third requirement starting from  $i$  and ending at  $j$ .

### Examples

stdin	stdout
4 5	1
1 2	3
3 4	4
1 3	1 2 3 4
2 3	
4 3	
1 4	

## Problem S. Minimum spanning tree

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 0.1 giây  
Memory Limit: 256 MB

The problem requires to find among the minimum spanning tree the minimum spanning tree with the smallest lexicographic order.

Definition the lexicographic order of a spanning tree:

- The spanning tree is represented by a list of edges of this spanning tree;
- Each edge is written by its two vertices in the order of the labels, *i.e.*:  $(u, v)$ ,  $u < v$
- The edges of the spanning tree is listed by the lexicographic order: the edge  $(u_1, v_1)$  is before the edge  $(u_2, v_2)$  if  $u_1 < u_2$  or  $(u_1 = u_2 \text{ and } v_1 < v_2)$ .

### Input

- The first line contains two positive integers  $n \leq 10^5$ ,  $m \leq 10^5$  that are the number of vertices and the number of edges of the given graph respectively;
- $m$  Each of the  $m$  following lines contains three positive integers  $u$ ,  $v$  and  $c$  corresponding to an edge  $(u, v)$  and its weight  $c$ ;

### Output

- The first line contains an integer that is the weight of the resulted spanning tree.
- The second line contains  $n - 1$  pairs of integers that are the edges of the resulted spanning tree, which are listed by the lexicographic order.

### Examples

stdin	stdout
4 6 2 1 3 2 3 1 3 1 10 4 1 5 3 4 5 4 2 6	9 1 2 1 4 2 3

## Problem T. The shortest path

Input File Name: `stdin`  
Output File Name: `stdout`  
Time Limit: 0.1 giây  
Memory Limit: 512 MB

The problem requires to find among the shortest paths of a given graph the shortest path with the smallest lexicographic order.

### Input

- The first line contains two positive integers  $n \leq 10^5$ ,  $m \leq 10^5$  that are the number of vertices and the number of edges respectively;
- Each of the following  $m$  lines contains three positive integers  $u$ ,  $v$  and  $c$  corresponding to the arc  $(u, v)$  and its weight  $c$ ;
- The last line contains two positive integers  $s$  and  $t$  that are the starting point and the ending vertex respectively of the required path.

### Output

- The first line contains an integer that is the weight of the resulted path.
- The second line contains the integers that are the vertices of the resulted path.

### Examples

stdin	stdout
4 6 2 1 3 2 3 1 3 1 10 4 1 5 3 4 5 4 2 6 2 4	6 2 3 4