Contents

$a + b = c$ — ADD 1
Largest weight subsequence — SUBSEQMAX 3
Sorting a sequence — SORT 4
Sorting a sequence by the combsort algorithm — COMBSORT 5
Checking parentheses expressions — PARENTHESES 6
Water jug problem — WATERJUG 7
Holey N-Queens — HOLEYQUEENS 8
Easy modified sudoku — EZSUDOKU 9
Bloper Operator — BLOPER 11
Wine trading in Gergovia — GERGOVIA 12
I am very busy — BUSYMAN 13
Aggressive cows — AGGRCOW 15
Eko — EKO 16
Copying Books — BOOKS1 17

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 real 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 $s = \langle a_1, \ldots, a_n \rangle$
- a subsequence is $s(i, j) = \langle a_i, \ldots, a_j \rangle$, $1 \leq i \leq j \leq n$
- 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 3 8 5 3 4 5	2 -1

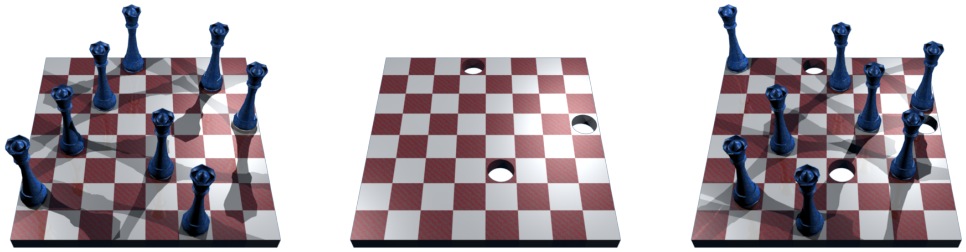
Scoring

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

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 9×9 grid with digits so that each column, each row, and each of the nine 3×3 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 3×3 subregions of the 9×9 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 8×8 grids and divided into 4 areas of 4×4 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 4×4 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 1 2 3 4 5 6 7 0 5 6 2 1 3 4 0 7 6 5 0 7 2 1 3 4 3 4 7 8 6 5 2 1 2 1 5 6 8 7 4 3 4 3 6 5 7 8 1 2 0 7 1 2 4 3 5 6 7 0 4 3 1 2 6 5 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 1 1 1 1 1 1 1 1	Test case #1: YES 1 2 3 4 5 6 7 8 5 6 2 1 3 4 8 7 6 5 8 7 2 1 3 4 3 4 7 8 6 5 2 1 2 1 5 6 8 7 4 3 4 3 6 5 7 8 1 2 8 7 1 2 4 3 5 6 7 8 4 3 1 2 6 5 Test case #2: NO

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 $A = \{1, 2, 3, \ldots, N\}$ 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 9 5 5 6	1 0

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 ith 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 ai 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.

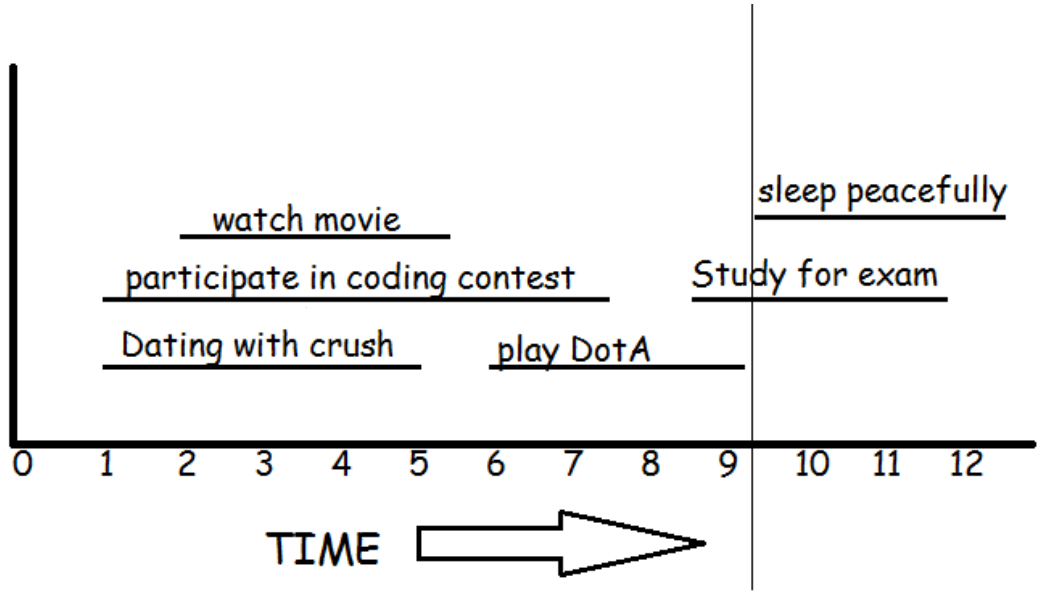
Examples

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

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

$1 \leq T \leq 10$

$1 \leq N \leq 100000$

$0 \leq start < end \leq 1000000$

Examples

Standard Input	Standard Output
3 3 3 9 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	1 2 3

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 $x_1, \ldots, x_N$ ($0 \leq x_i \leq 1,000,000,000$).

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 $2 \ldots N+1$: 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 scribers. The scriber had been given a book and after several months he finished its copy. One of the most famous scribers 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 scribers.

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 scribers to make copies of these books. Imagine you have m books (numbered $1, 2, \ldots, m$) that may have different number of pages $(p_1, p_2, \ldots, p_m)$ 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 scriber only, and every scriber must get a continuous sequence of books. That means, there exists an increasing succession of numbers $0 = b_0 < b_1 < b_2, \ldots < b_{k-1} \leq b_k = m$ such that i-th scriber 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 scriber who was assigned the most work. Therefore, our goal is to minimize the maximum number of pages assigned to a single scriber. 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 $p_1, p_2, \ldots, p_m$ 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 $p_1, p_2, \ldots, p_m$ 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 scriber, then to the second scriber etc. But each scriber must be assigned at least one book.

Examples

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