Problem A. Toby the adventurer

Source file name:	adventurerd2.c, adventurerd2.cpp, adventurerd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Manuel Felipe Pineda - UTP Colombia`

Toby is a great adventurer. Today he is trying to explore "Bitland" (a new country that will be remembered after Toby's exploration).

Bitland is divided into N small cities and M unidirectional roads between cities.

Toby begins the adventure at the city R, and after that he goes to any city R', if this new city (R') is not known by Toby, a road between R and R' is needed and he must pay a cost (in terms of adventure power) associated to the road. Otherwise, if Toby wants to go to a known city he does not need pay anything, even if there is no road from the current city to the target city (like teleportation)... is not Toby so cool?

Toby keeps traveling between cities until he reaches every city in Bitland. After this moment Toby goes to home, happy and eager for new adventures.

Wait! Where is the problem?

Did you remember that Toby has to pay for each road that is used to disclose a new city? Help Toby to minimize this cost (the sum of all power paid), because he needs as much energy as possible for his new adventures.

Input

The input starts with an integer $1 < T \leq 100$ indicating the number of test cases.

Each test case begins with three integers $3 < N \leq 10\,000$, $3 < M \leq N$, $0 \leq R < N$ denoting the number of cities, number of roads and initial city, respectively. Followed by M lines which contain three integers, $0 \leq u, v < N$, $1 \leq w \leq 10\,000$. These numbers denote a road from the city u to the city v with cost w.

Note that there could be several roads between the same pair of cities

Output

Print one line with the total cost for the adventure, followed by $N - 1$ lines with the chosen roads in the same format that was given in the input:

u v w - three space separated integers denoting a road from u to v with cost w.

If there are several answers, print any of them.

If there is no way to visit all the N cities, print "impossible" without quotes.

Example

Input	Output
3 5 5 0 0 1 1 0 2 100 1 3 2 3 2 3 2 4 4 5 5 4 0 1 1 0 2 100 1 3 2 3 2 3 2 4 4 4 4 0 0 1 3 0 2 4 3 1 1 2 3 1	10 0 1 1 3 2 3 1 3 2 2 4 4 impossible 6 3 1 1 0 2 4 2 3 1

Use faster I/O methods

Problem B. The book thief

Source file name:	bookd2.c, bookd2.cpp, bookd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

On February 18, 2014, Red Matemática proposed the following mathematical challenge on their twitter account (@redmatematicant): "While Anita read: *The book thief* by Markus Zusak, She added all the page numbers starting from 1. When she finished the book, she got a sum equal to 9.000 but she realized that one page number was forgotten in the process. What is such number? and, how many pages does the book have?"

Using this interesting puzzle as our starting point, the problem you are asked to solve now is: Given a positive integer s ($1 \leq s \leq 10^8$) representing the result obtained by Anita, find out the number of the forgotten page and the total number of pages in the book.

Input

The input may contain several test cases. Each test case is presented on a single line, and contains one positive integer s. The input ends with a test case in which s is zero, and this case must not be processed.

Output

For each test case, your program must print two positive integers, separated by a space, denoting the number of the forgotten page and the total number pages in the book. Each valid test case must generate just one output line.

Example

Input	Output
1	2 2
2	1 2
3	3 3
4	2 3
5	1 3
6	4 4
9000	45 134
499977	523 1000
49999775	5225 10000
0	

Use faster I/O methods

Problem C. Numeric Center

Source file name:	centerd2.c, centerd2.cpp, centerd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Hugo Morales, Sebastián Gómez & Santiago Gutierrez - UTP Colombia`

A numeric center is a number that separates in a consecutive and positive integer number list (starting at one) in two groups of consecutive and positive integer numbers, in which their sum is the same. The first numeric center is number 6, which takes the list $\{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8\}$ and produces two lists of consecutive and positive integer numbers in which their sum (in this case 15) is the same. Those lists are: $\{1,\ 2,\ 3,\ 4,\ 5\}$ and $\{7,\ 8\}$. The second numeric center is 35, that takes the list $\{1,\ 2,\ 3,\ 4, \ldots,\ 49\}$ and produces the following two lists: $\{1,\ 2,\ 3,\ 4, \ldots,\ 34\}$ and $\{36,\ 37,\ 38,\ 39, \ldots,\ 49\}$, the sum of each list is equal to 595.

The task consists in writing a program that calculates the total of numeric centers between 1 and n.

Input

The input consists of several test cases. There is only one line for each test case. This line contains a positive integer number n ($1 \leq n \leq 10^{14}$). The last test case is a value of n equal to zero, this test case should not be processed.

Output

For each test case you have to print in one line, the number of numeric centers between 1 and n.

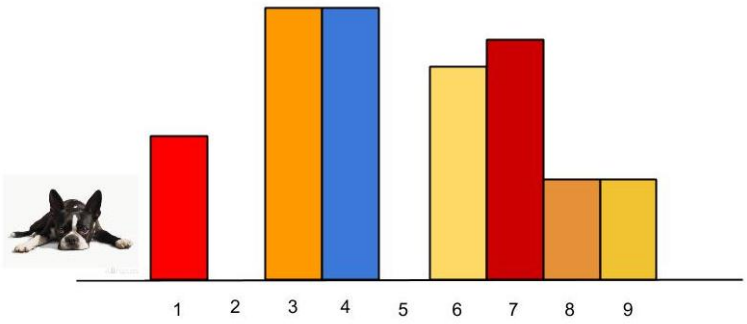
Example

Input	Output
1	0
7	0
8	1
48	1
49	2
50	2
0	

Problem D. Toby and the playground

Source file name:	playgroundd2.c, playgroundd2.cpp, playgroundd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan David Gil López - UTP Colombia`

Toby likes to play a lot, as well as he likes to be a little lazy. He has a special room with lots of stuff for his entertainment. One of these peculiar toys, is a set of arranged columns (like in the image below) that he would like to climb up and down, but, given the fact that there are some missing columns, Toby will not agree to make more effort jumping over separated columns, and for that reason you have to put new columns in order to complete the playground, that is, put new columns in all the empty spaces.



You know that even though Toby is playful he is also lazy, so, to help him, you go to a shop to buy new columns of different sizes to complete Toby's playground, but you don't want Toby to get very tired playing, so you want to arrange the new columns in such a way that minimizes an energy wasting function. This function is given as follows: $\sum_{i=2}^{n} |h_i - h_{i-1}|$ where n is the amount of columns that Toby will pass through and h_i is the size of a column i.

The specific task you will have to solve in order to please Toby is: Given an arrangement of n columns $h_1, h_2, h_i, \ldots, h_n$ and m new possible column sizes sizes $H_1, H_2, \ldots, H_i, \ldots, H_m$, you have to minimize the energy wasting function given above. You can assume that the shop offers an 'infinite' amount of columns of size H_i, also, take into account that there could be a lot of consecutive empty spaces.

Input

The input will consist of several test cases. In the first line are given two integers n ($1 \leq n \leq 100\,000$) and m ($1 \leq m \leq 200\,000$) representing respectively, the number of columns in the arrangement and the amount of columns sizes that you can use to complete the playground. Then, in the second line, n integers $h_1, h_2, \ldots, h_i, \ldots, h_n$ are given, where h_i ($0 \leq h_i \leq 200\,000$) represents the size of column i, a column which it's size is 0 represents an empty space. It's ensured that there is at least one empty space. In the third and last line, you will be given m integers $H_1, H_2, \leq, H_i, \leq, Hm$ ($0 \leq H_i \leq 200\,000$) representing the sizes of several types of columns offered in the shop, no two sizes of columns offered in the shop are the same. The end of the test cases is given by the end of file (EOF).

Output

The output should be a single integer, the minimum amount of total energy wasted by Toby using the function given. Each test case must generate just one output line.

Example

Input	Output
2 3 10 0 1 5 20 3 4 8 0 5 2 6 7 12	5 3

Use faster I/O methods

Problem E. Toby and the frog

Source file name:	frogd2.c, frogd2.cpp, frogd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Manuel Felipe Pineda - UTP Colombia`

Toby the dog is on the cell 0 of a numbered road, TJ the frog is on the cell number X of the same road. Toby wants to catch the frog, but he is not smart enough to count the distance from his current position to X, so he performs the following algorithm:

- Let *pos* be the Toby's current position.
- Jump a distance d, d is uniformly distributed over $[1, min(X - pos,\ 10)]$.
- If the new position is the frog's position, catch it and send it as tribute to the queen.
- In other case start the algorithm again.

Note that the length of Toby's jump cannot be infinite, in fact, it must be less than or equal to 10. Besides this, he will never jump over the frog, in other words, he will never reach a position greater than X.

TJ the frog does not want to be catched, due to this, TJ wants to compute the expected number of jumps that Toby needs in order to reach cell number X.

Help to TJ compute this value.

Input

The input starts with an integer $1 < T \leq 100$ indicating the number of test cases.

Each test case contains one integer $10 \leq X \leq 5000$ denoting the frog's cell

Output

For each test case print in one line the expected number of jumps that Toby needs to reach cell number X.

Answers with relative error less than 10^{-6} will be considered correct.

Example

Input	Output
2 10 20	2.9289682540 4.8740191199

Problem F. Toby and the strange function

Source file name:	stranged2.c, stranged2.cpp, stranged2.java
Input:	Standard
Output:	Standard
Author(s):	Jhon Jimenez - UTP Colombia

As is well known, Toby is a cute and smart dog, but this problem is too hard even for Toby. For this problem he needs to find a function f that receive two arguments, an integer n and a string S and return a string S', more formally $f(n, S) = S$'.

Input

The first line contains a single integer T denoting the number of test cases. Each case in the first line contains an integer n ($0 \leq n \leq 10^{18}$) and the second line contains S (the string only contains lowercase Latin letters), the length of S does not exceed 100 characters.

Output

For each test case you have to print in one line the string S', value of $f(n,\ S)$.

Example

Input	Output
3 1 abcd 2 abcd 4 abcd	dabc cdab abcd

Explication

$f(1,\ abcd) = dabc$
$f(2,\ abcd) = cdab$
$f(4,\ abcd) = abcd$

Can you help the poor dog in this complicated task?

Problem G. Grounded

Source file name:	groundedd2.c, groundedd2.cpp, groundedd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Sebastián Gómez - UTP Colombia`

Toby was behaving badly at little dog school and his teacher grounded him by asking him to solve a hard problem. Toby is given a number N, let's consider a set S of all binary strings of N bits. Let's also consider any subset P_i of S, let $XOR(P_i)$ be the XOR of all the elements of P_i. The XOR of the empty set is a binary string of N zeros.

As Toby is a very smart dog, and Toby's teacher wants Toby to spend a very long time working on the problem, he asks:

How many different subsets P_i of S exist such than $XOR(P_i)$ has exactly K ones?

Recall that the empty set and S itself are valid subsets of S.

Input

The input consist of several test cases. Each test case consists of a line containing the numbers N and K. The end of the test cases is given by the end of file (EOF).

- $1 \leq N \leq 4$
- $0 \leq K \leq N$

Output

For each test case print in one line the requested answer.

Example

Input	Output
2 0 1 1	4 2

Explication

For the first test case the subsets of the strings of 2 bits with an XOR with zero ones is: {}, {00}, {01, 10, 11} and {00, 01, 10, 11}

For the second test case the subsets of the strings of 1 bit with an XOR with one is: {1}, {0, 1}

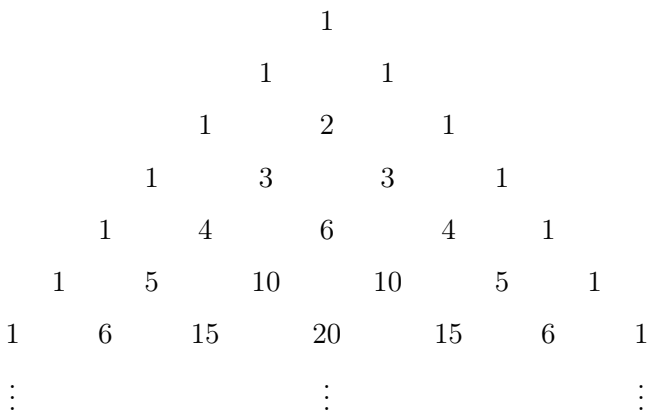
Problem H. Pascal's triangle: Sum of levels.

Source file name:	pascald2.c, pascald2.cpp, pascald2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Hugo Humberto Morales Peña - UTP Colombia`

From Wikipedia, the free encyclopedia : "In mathematics, Pascal's triangle is a triangular array of the binomial coefficients. In much of the Western world it is named after French mathematician Blaise Pascal, although other mathematicians studied it centuries before him in India, Iran, China, Germany, and Italy."

Pascal's triangle is a triangle of integers, infinite and symmetrical, begins with a 1 in the first row (zero level triangle), and in the following rows are placed numbers in such way that each one is the sum of the two numbers it has above. It is assumed that the places outside the triangle contain zero, so that the edges of the triangle are formed by ones.

The following image contains the first levels of Pascal's triangle (from level 0 to level 6):



In this problem you must compute the sum of all of Pascal's triangle's elements from level m to level n.

Input

The input may contain several test cases. Each test case is presented in a single line, and contains two positive integers m and n ($0 \leq m \leq n \leq 1000$). The input ends with $m = -1$ and $n = -1$, and this case must not be processed.

Output

For each test case, your program must print the sum of all Pascal's triangle's elements from level m to level n mod 1 000 007. Each valid test case must generate just one output line. *mod operation means the remainder after dividing of one number by* 1 000 007. *It is denoted by the operator % in several programming languages.*

Example

Input	Output
0 0 1 2 1 3 4 6 -1 -1	1 6 14 112

Use faster I/O methods

Problem I. Toby and the Graph

Source file name:	tobygraphd2.c, tobygraphd2.cpp, tobygraphd2.java
Input:	`Standard`
Output:	`Standard`
Author(s):	`Jhon Jimenez - UTP Colombia`

Toby has an undirected graph where not necessarily all vertices are connected to each other. As a result, we could have some disjoint sets of interconnected vertices. Toby is a curious and smart dog, so he is wondering about the following question: how many new sets can we get after joining **any two** initial sets?

Input

The first line contains a single integer T denoting the number of test cases. Each case in the first line contains two integers n ($1 \leq n \leq 10^4$) and m ($0 \leq m \leq n-1$), the number of vertices and the number of edges respectively. The next m lines contains two integers separated by a single space a, b ($1 \leq a \leq b \leq n$) meaning that vertex a is connected to vertex b. Each vertex is enumerated from 1 to n.

Output

Print the answer in a separate line.

Example

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

Explication

First input:

- Initially we have 4 sets: $A = \{1\}$, $B = \{7\}$, $C = \{2, 3\}$, $D = \{4, 5, 6\}$.
- We can get 6 new sets: $A \cup B = \{1, 7\}$, $A \cup C = \{1, 2, 3\}$, $A \cup D = \{1, 4, 5, 6\}$, $B \cup C = \{2, 3, 7\}$, $B \cup D = \{4, 5, 6, 7\}$, $C \cup D = \{2, 3, 4, 5, 6\}$.
- Toby cares about the number of new sets so the answer is 6.

Second input:

- Initially we have 1 set: $A = \{1\}$.
- It is impossible to get any new sets, so the answer is 0.

Problem J. Josephus lottery

Source file name:	josephusd2.c, josephusd2.cpp, josephusd2.java
Input:	Standard
Output:	Standard
Author(s):	Hugo Humberto Morales Peña - UTP Colombia

Professor Humbertov Moralov wants to make a raffle between the students of his Data Structure class and Pepito (a student of this group) suggests to use the Josephus problem to determine who is the winner of the raffle. The problem is that you can know beforehand the winning position if you know the value of n (the total of students in the raffle) and the value k (the amount of movements before throwing out a student from the circle).

The prize is kind of interesting, the winner won't have to take the final exam, and for that reason the professor Humbertov proposes the following variant to the Josephus problem: "Take the student class list, in which the students are numbered from 1 to n, then, organize these numbers in a circle and begin to count clockwise from number 1 to the value k. The student with number k in the list is removed from the circle, and now you begin to count, now counterclockwise, from the number of the next student $(k+1)$. The student with the number in which the count stopped is removed from the circle, and then you repeat the process alternating between clockwise and counterclockwise, counting until you get the winner of the raffle".

Input

The input contains several test cases. Each test case has only one line, in which there are two positive integers N $(1 \leq N \leq 10^4)$ and K $(1 \leq K \leq N)$ that represents respectively, the number of students in the raffle and the value of movements to remove students from the circle. The input ends with a case containing two zeros, which must not be processed.

Output

For each test case you have to print in one line, the number in the student list that represents the winner of the raffle.

Example

Input	Output
10 1	6
10 5	2
10 10	5
5 5	4
5 4	2
0 0	

Explication

This is the sequence for each step in the case "5 4": 1 2 3 4 5
1 2 3 ~~4~~ 5
1 2 3 5
~~1~~ 2 3 5
2 3 5
2 3 ~~5~~
2 3
2 ~~3~~
2 ← The winner