Problem A. Toby the adventurer

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

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>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</pre></td><td><pre>10
0 1 1
3 2 3
1 3 2
2 4 4
impossible
6
3 1 1
0 2 4
2 3 1</pre></td></tr>
</table>

Use faster I/O methods

Problem B. The book thief

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

Use faster I/O methods

Problem C. Numeric Center

Input: `Standard`
Output: `Standard`
Author(s): `Hugo H. Morales, Sebastián Gómez and 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 395.

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

Problem D. Snakes and Ladders

Input: `Standard`
Output: `Standard`
Author(s): `Sebastián Gómez - UTP Colombia`

Snakes and ladders is a popular game for kids (and cute Dogs of course). Usually this game is played between multiple players but Toby does not like the other pups in his school, and wants to play alone. The game is very simple, Toby starts at position 1 of a board of height H and width W and the goal is to get to position $H \times W$.

Each turn Toby rolls a fair die and advances a number of positions equal to the result of the die. If at the end of a turn Toby lands at the bottom of a ladder he advances immediately to the top, and if Toby lands at the head of a snake then he goes back to the tail of the snake immediately as well.

Board of the third test case sample

Remember that a fair die is a die where the probability to get any outcome between 1 and 6 is the same. In the figure 1 you can see a sample board. To explain what happens when Toby is close to the finish let's make an example with this board. Let's suppose that Toby is at position 29. Then Toby rolls the die, if he gets one he advances to position 30 and wins. If he gets 2, he lands in 29 again (Advance one and go one back). If he gets, 3 he lands in 28 (Advance one and go two back). If he gets 4, he lands in 27 and then immediately goes to position 1 since he stepped in the head of a snake.

Now Toby wants to know how long will it take his game before it ends, and he asks you to compute the expected amount of turns (die rolls) before he wins. It is guaranteed that it is always possible to reach the goal of the board and that the maximum expected number of turns will not exceed 100 000. The starting cell will never be the base of a ladder and the target cell will never be the head of a snake.

Input

The input consists of several test cases. Each test case begins with a line with three integers W,H and S. Here W and H are as above and S is the number of snakes or ladders. Then follow S lines, each with two integers u_i and v_i meaning if you land in the cell u_i you have to go to cell v_i immediately. So if $u_i < v_i$ it is a ladder and if $u_i > v_i$ it is a snake. It is guaranteed that $u_i \neq u_j \forall i \neq j$ and $u_i \neq v_j \forall i, j$. Read input until end of file is reached, there will be a blank line after each test case.

- $1 \leq W, H \leq 12$
- $W \times H \geq 7$
- $0 \leq S \leq \frac{W \times H}{2}$
- $1 \leq u_i, v_i \leq W \times H$

Output

For each test case print a single number consisting on the expected number of turns to finish the game. The answer will be considered correct if the difference with respect to the right answer is less than 10^{-2}.

Example

Input	Output
7 1 0 6 5 0 6 5 8 3 22 17 4 5 8 19 7 21 9 11 26 27 1 20 29	6.00000000 13.04772792 19.83332560

Problem E. Subset sum

Input:	`Standard`
Output:	`Standard`
Author(s):	`Sebastián Gómez - UTP Colombia`

Given a set s of integers, your task is to determine how many different non-empty subsets sum up to a target value.

Input

The input consists of several test cases. The first line of each test case is a line containing two integers N and T, the number of items of the original set of integers and the target value. After that comes one line with the N integers s_i that belong to the original set s.

- $1 \leq N \leq 40$
- $-10^9 \leq T,\ s_i \leq 10^9$

Output

For each test case print on a single line an integer indicating the number of different non-empty subsets that sum up to the target value T.

Example

Input	Output
6 0 -1 2 -3 4 -5 6 5 0 -7 -3 -2 5 8	4 1

Explication

On the first test case the target is 0 and the following are the valid subsets: (2, 4, -1, -5), (2, 6, -5, -3), (4, -1, -3), (6, -5, -1). On the second test case the target is again 0, the only valid subset is: (-3, -2, 5)

Problem F. Josephus lottery

Input:	`Standard`
Output:	`Standard`
Author(s):	`Hugo Humberto Morales Peña and Sebastián Gómez - 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 $(1 \leq N \leq 10^6)$ 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 the number in the student list that represents the winner of the raffle.

Example

Input	Output

`10 1`
`10 5`
`10 10`
`5 5`
`5 4`
`0 0`
	`6`
	`2`
	`5`
	`4`
	`2`

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

Problem G. Grounded

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).

- $1 \leq K \leq N \leq 10^6$

Output

For each test case print the requested answer modulo $p = 10^9 + 7$.

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. Sum of all permutations

Input: `Standard`
Output: `Standard`
Author(s): `Sebastián Gómez - UTP Colombia`

Toby is very bored because his father went to live to Brazil, so he decided to create a challenge that might take a lot of time to solve. First he creates a function called

SadToby

that receives an array of integers called permutation and a number M as follows:

```
def SadToby(permutation, M):
    sum = 0
    for each x in permutation:
        if (x <= M):
            sum = sum + x
        else:
            break
    return sum
```

For every permutation of the numbers from 1 to N Toby needs to print the sum of SadToby function. Toby needs to compute this result for every possible value of M between 1 and N. As each of this values can be very large output the result modulo the prime $p = 1711276033 = 2^{25} \times 51 + 1$. Can you help this cute dog with his task?

Input

The input consists of several test cases. Each test case begins with a line with one integers N.

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

Output

For each test case, print a single line with N integers containing the required sum for every value of M between 1 and N.

Example

Input	Output
1 2 3	1 1 6 2 9 36

Explication

Third case, first output number $M = 1$. Consider all permutations. If the first number is greater than 1, then the loop will break in the beginning itself with output 0. There are a total of 6 distinct permutations out of which 4 will give 0. The remaining 2 will fetch 1 each from the function. Thus the answer is 2. For $M = 2$ it's easy to check that the output is 9 and for $M = 3$ is 36.

Problem I. TripleCorn

Input:	`Standard`
Output:	`Standard`
Author(s):	`Sebastián Gómez - UTP Colombia`

TripleCorn is a company that produces corn cobs triple as big and double as tasty as any other company. Toby, the cute, small and smart dog, has been hired by Triplecorn to carry Triplecobs between cob baskets.

The manager has initialy N Triplecobs and a line of baskets numbered from 1 to 10000. Each of the Triplecobs is in one of the baskets. The manager wants toby to group the Triplecobs on K baskets or less in a way he minimices the amount of tobycookies he must give to Toby. Toby is a small dog, and can only carry one TripleCob at a time. So Toby must do his job by picking a Corncob on some basket, walking to some other basket and releasing the Triplecob there. The reward for Toby is a tobycookie for every meter that he walks with one of those taisty cobs on his mouth (Without eating it). The distance between two consecutive baskets in one meter. Remember toby doesn't charge if he walks without a Triplecorn on his mouth.

Input

The input consists of several test cases. Each test case begins with a line with two integers N and K. Then follows one line with N integers C_i indicating the initial basket of the Triplecob i.

- $1 \leq N \leq 10000$
- $1 \leq K \leq N$
- $1 \leq C_i \leq 10000$

Output

For each test case, print a single line containing the minimum amount of tobycookies the manager must pay Toby to do the job.

Example

Input	Output

`3 3`
`1 2 3`
`3 1`
`1 2 3`
`4 2`
`1 3 8 9`
	`0`
	`2`
	`3`

Explication

In the first test case the objects are already grouped on 3 baskets, so Toby doesn't need to do any job. In the second test case, Toby can move the cobs in baskets 1 and 3 to basket 2 with cost 1 each (Total cost 2), note that there is no better solution.