1. Airport

Lolek and Bolek are about to travel abroad by plane. The local airport has a special "Choose Your Plane" offer. The offer's conditions are as follows:

1. it is up to a passenger to choose a plane to fly on;
2. if the chosen plane has x ($x > 0$) empty seats at the given moment, then the ticket for such a plane costs x zlotys (units of Polish currency).

The only ticket office of the airport already has a queue of n passengers in front of it. Lolek and Bolek have not stood in the queue yet, but they are already wondering what is the maximum and the minimum number of zlotys the airport administration can earn if all npassengers buy tickets according to the conditions of this offer?

The passengers buy tickets in turn, the first person in the queue goes first, then goes the second one, and so on up to n-th person.

Input

The first line contains two integers n and m ($1 \le n, m \le 1000$) — the number of passengers in the queue and the number of planes in the airport, correspondingly. The next line contains m integers a1, a2, ……an (1<=ai<=1000) - ai stands for the number of empty seats in the i-th plane before the ticket office starts selling tickets.

The numbers in the lines are separated by a space. It is guaranteed that there are at least n empty seats in total.

Output

Print two integers - the maximum and the minimum number of zlotys that the airport administration can earn, correspondingly.

Examples

Input 1

4 3
2 1 1

Output 1

5 5

Input 2

4 3
2 2 2

Output 2

7 6

Note

In the first test sample the number of passengers is equal to the number of empty seats, so regardless of the way the planes are chosen, the administration will earn the same sum.

In the second sample the sum is maximized if the 1-st person in the queue buys a ticket to the 1-st plane, the 2-nd person — to the 2-nd plane, the 3-rd person — to the 3-rd plane, the 4-th person — to the 1-st plane. The sum is minimized if the 1-st person in the queue buys a ticket to the 1-st plane, the 2-nd person — to the 1-st plane, the 3-rd person — to the 2-nd plane, the 4-th person — to the 2-nd plane.

Testcase:

40 10

1. 2 3 4 5 6 7 10 10 10

2. Just another question

Lorrie and Matt are best friends. They spend a lot of time together finding about prime numbers and their usage, and powers. Now, they've managed to find out about a new type of number called the: dual numbers.

> The dual numbers are special numbers which are produced when two prime numbers are selected and multiplied. For example, 4 is a dual number, since it is the product of 2 prime numbers, 2 and 2. Similarly, 6 is also a dual number, since it is the product of 2 and 3. But, 8 is not a dual number, since it is not the product of any two prime numbers.

Now Lorrie and Matt are given an array consisting of N numbers. The numbers can be arbitrary positive integers. Lorrie and Matt will each independently choose a prime number in the array (their choice of numbers does not necessarily have to be distinct). Then, they will take their choices and multiply them to form a dual number. You're supposed to help them out by finding out the maximum dual number which they can create from this process. If any dual number cannot be generated from the given array, output -1.

Input format:

The first line consists of a number t, which denotes the number of test cases. For every test case, there will be two lines. On the first line, there will be a number n, denoting the size of the array. On the second line will be n positive integers a1,a2,...,an separated by a space.

Output format:

For every test case, you have to print the maximum dual number which can be formed by the given array. In case this is NOTpossible, print -1.

Constraints:

1<=t<=100

1<=n<=10^3

1<=ai<=10^3

Sample Input:

1

5

1 4 6 8 10

Sample Output:

-1

Test Case:

1

12

1 2 56 125 34 123 56 12 2 1 23 125

3. Killing peeps

The fictional university of ZombieLand - has n students volunteer for the role of Killing Assistants, student volunteers who help in smooth killing of zombies in the campus. Every student can either manage to shoot a zombie or miss it.

Out of these n Killing Assistants, m students try to shoot the zombie fairly, that is, they are unbiased, while the rest of them are biased. The probability that a zombie will be hit by a shot if the Killing Assistant is biased is 2/3, otherwise the probability is 1/2.

Little Harry figures out this mess by selecting a Killing Assistant randomly. He tricks that particular Killing Assistant in shooting the same zombie twice. The first time, the Killing Assistant shoots the zombie, while the other time, he ends up missing it.

Harry wants to know what are the chances that the selected Killing Assistant was NOT biased.

Input format:

The first line consists of a number t, which denotes the number of test cases. For every test case, there will be two integers on the next line. The number n denoting the number of students, while the number m denotes the number of unbiased students.

Output format:

For every test case, print the required probability in the form of a reduced fraction possible. More formally, print the desired probability as P/Q, where P and Q are coprime.

Constraints:

1<= t <= 10^5

1 <= n <=10^5

0 <= m <= N

SAMPLE INPUT

3

2 1

8 5

3 3

SAMPLE OUTPUT

9/17 15/23 1/1

Testcase:

5

11 9

9 9

12 3

4 1

29 13

4. Playing Cubes

Petya and Vasya decided to play a little. They found n red cubes and m blue cubes. The game goes like that: the players take turns to choose a cube of some color (red or blue) and put it in a line from left to right (overall the line will have n + m cubes). Petya moves first. Petya's task is to get as many pairs of neighbouring cubes of the same color as possible. Vasya's task is to get as many pairs of neighbouring cubes of different colors as possible.

The number of Petya's points in the game is the number of pairs of neighboring cubes of the same color in the line, the number of Vasya's points in the game is the number of neighbouring cubes of the different color in the line. Your task is to calculate the score at the end of the game (Petya's and Vasya's points, correspondingly), if both boys are playing optimally well. To "play optimally well" first of all means to maximize the number of one's points, and second - to minimize the number of the opponent's points.

Input

The only line contains two space-separated integers n and m (1<=n, m<=10^5) - the number of red and blue cubes, correspondingly.

Output

On a single line print two space-separated integers — the number of Petya's and Vasya's points correspondingly provided that both players play optimally well.

Examples

Input 1

3 1

Output 1

2 1

Input 2

2 4

Output 2

3 2

Note

In the first test sample the optimal strategy for Petya is to put the blue cube in the line. After that there will be only red cubes left, so by the end of the game the line of cubes from left to right will look as [blue, red, red, red]. So, Petya gets 2 points and Vasya gets 1 point.

If Petya would choose the red cube during his first move, then, provided that both boys play optimally well, Petya would get 1 point and Vasya would get 2 points.

Testcase:

5 13

5. XOR

You are given an array A1, A2,...,An. You have to tell how many pairs **(i, j)** exist such that **1 <= i < j <= N** and **Ai XOR Aj** is **odd**.

Input and Output
First line **T**, the number of testcases. Each testcase: first line **N**, followed by **N** integers in next line. For each testcase, print the required answer in one line.

Constraints
1 <=T <= 10
1 <= N <= 10^5
0 <= Ai <= 10^9

Sample Input:
2
3
1 2 3
4
1 2 3 4

Sample Output:
2 4

Test Case:

2

20

1 2 3 4 5 5 4 3 2 1 233 576 8901 9123 912 233 77 900 123 20

12

1 43 213 543 6 14 53 90 0 32 1 43

6. Robot Attack

Vasya plays Robot Bicorn Attack.

The game consists of three rounds. For each one a non-negative integer amount of points is given. The result of the game is the sum of obtained points. Vasya has already played three rounds and wrote obtained points one by one (without leading zeros) into the string s. Vasya decided to brag about his achievement to the friends. However, he has forgotten how many points he got for each round. The only thing he remembers is the string s.

Help Vasya to find out what is the maximum amount of points he could get. Take into account that Vasya played Robot Bicorn Attack for the first time, so he could not get more than 1000000 points for one round.

Input

The only line of input contains non-empty string s obtained by Vasya. The string consists of digits only. The string length does not exceed 30 characters.

Output

Print the only number - the maximum amount of points Vasya could get. If Vasya is wrong and the string could not be obtained according to the rules then output number -1.

Examples

Input 1

1234

Output 1

37

Input 2

0009

Output 2

-1

Note

In the first example the string must be split into numbers 1, 2 and 34.

In the second example the string is incorrect, because after splitting the string into 3 numbers number 00 or 09 will be obtained, but numbers cannot have leading zeroes.

Testcase:

69284626624

7. Sleep in Class

The academic year has just begun, but lessons and hackathons have already occupied all the free time.

It is not a surprise that today Walia fell asleep in the EMW class(Electromagnetic Waves, silly). He had a dream in which he was on the stairs.

The stairs consists of n steps. The steps are numbered from bottom to top, it means that the lowest step has number 1, and the highest step has number n.

Above each of them there is a pointer with the direction (up or down) Walia should move from this step.

As soon as Walia goes to the next step, the direction of the pointer (above the step he leaves) changes.

It means that the direction "up" changes to "down", the direction "down" - to the direction "up".

Walia always moves to the next step in the direction which is shown on the pointer above the step.

If Walia moves beyond the stairs, he will fall and wake up. Moving beyond the stairs is a moving down from the first step or moving up from the last one (it means the n-th) step.

In one second Walia moves one step up or down according to the direction of the pointer which is located above the step on which Walia had been at the beginning of the second.

For each step find the duration of the dream if Walia was at this step at the beginning of the dream.

Walia's fall also takes one second, so if he was on the first step and went down, he would wake up in the next second.

Input

The first line contains single integer n (1<=n<=10^6) - the number of steps on the stairs.

The second line contains a string s with the length n - it denotes the initial direction of pointers on the stairs.

The i-th character of string s denotes the direction of the pointer above i-th step, and is either 'U' (it means that this pointer is directed up),

or 'D' (it means this pointed is directed down).

The pointers are given in order from bottom to top.

Output

Print n numbers, the i-th of which is equal either to the duration of Walia's dream or to -1 if Walia never goes beyond the stairs, if in the beginning of sleep he was on the i-th step.

Sample Input

3

UUD

Sample Output

5 6 3

Test Case:

50

DDDDDDDDDDDDDDDDDDDDDDDDDUUUUUUUUUUUUUUUUUUUUUUUUU

8. Cipher

Jack and Daniel are friends. They want to encrypt their conversation so that they can save themselves from interception by a detective agency.

So they invent a new cipher.

Every message is encoded to its binary representation B of length N.

Then it is written down K times, shifted by 0,1,...,K-1 bits.

If B=1001010 and K=4 it looks so:

```
1001010
1001010
 1001010
  1001010
```

Then calculate XOR in every column and write it down. This number is called S.

For example, XOR-ing the numbers in the above example results in

1110100110

Then the encoded message S and K are sent to Daniel.

Jack is using this encoding algorithm and asks Daniel to implement a decoding algorithm.

Can you help Daniel implement this?

Input Format

The first line contains two integers N and K.

The second line contains string of length consisting of ones and zeros.

Output Format

Decoded message of length N, consisting of ones and zeros.

Constraints

1<=N<=10^6

1<=K<=10^6

|S|=N+K-1

It is guaranteed that S is correct.

Sample Input 0

7 4

1110100110

Sample Output 0

1001010

Explanation

1001010

1001010

1001010

1001010

1110100110

Test case:

185 48
0101001101011000110011010101000111011011111110001001110101111011010001000110
1100101000101011011000000010011011001100101110010101110101010101100110111001
0100010000000111111000010100110101110100010001010111101001011001010010101 1100
1001

Note* - All on one line

9. Closest Pair

You are given a weighted graph with N nodes and M edges. Some of the nodes are marked as *special* nodes. Your task is to find the shortest pairwise distance between any two different *special* nodes.

Input

The first line of the input contains three space-separated integers N, M and K denoting the number of nodes, the number of edges, and the number of special nodes.

The following line contains K space-separated distinct integers A1, A2, ..., AK , denoting the special nodes.

Each of the following M lines (say, the jth) contains a triple Xj Yj Zj, denoting the edge connecting the nodes Xj and Yj, and having the weight of Zj.

Output

Output the shortest pairwise distance between any two different special nodes.

Constraints

2 <= K <= N

The given graph is connected.

The given graph doesn't contain self-loops and multiple edges.

1 <= Ai <= N

1 <= Zj <= 10^4

1 <= Xj, Yj <= N

Input:

5 5 3

1 3 5

1 2 3

2 3 4

3 4 1

4 5 8

1 5 19

Output:

7

Explanation

Nodes 1, 3 and 5 are special nodes. Shortest distance between nodes 1 and 3 is 7 and that between nodes 3 and 5 is 9. Shortest distance between nodes 1 and 5 is 16. Minimum of these distances is 7. Hence answer is 7.

Test case:

13 15 4

7 9 11 13

1 11 15

1 9 60

3 8 54

3 6 95

4 9 74

5 7 52

5 10 35

7 3 79

8 6 84

3 2 39

2 1 15

6 12 72

10 1 30

12 4 42

8 13 104

10. Lucky Tickets

You are given a number 1 <=N <= 50. Every ticket has its 2N-digit number. We call a ticket lucky, if the sum of its first N digits is equal to the sum of its last N digits. You are also given the sum of ALL digits in the number. Your task is to count an amount of lucky numbers, having the specified sum of ALL digits.

INPUT:

Two space-separated numbers: N and S. Here S is the sum of all digits. Assume that 0 <= S <= 1000.

OUTPUT:

The amount of lucky tickets.

SAMPLE INPUT:

2 2

SAMPLE OUTPUT:

4

The tickets are 0101, 0110, 1001, 1010 in the example above

Testcase:

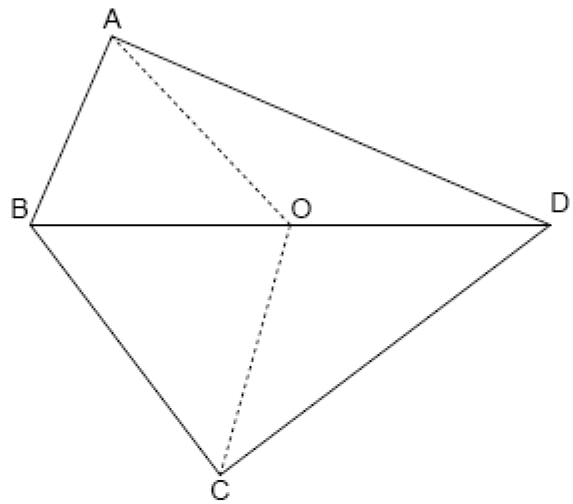
3 6

11. Biclinic Integral Quadrilaterals

ABCD is a convex, integer sided quadrilateral with 1 <= AB < BC < CD < AD.
BD has integer length. O is the midpoint of BD. AO has integer length.
We'll call ABCD a biclinic integral quadrilateral if AO = CO <= BO = DO.

For example, the following quadrilateral is a biclinic integral quadrilateral:
AB = 19, BC = 29, CD = 37, AD = 43, BD = 48 and AO = CO = 23.

Let B(N) be the number of distinct biclinic integral quadrilaterals ABCD that satisfy
AB^2+BC^2+CD^2+AD^2 <= N



Testcase :

10000

12. Lift and Throw

You are given a straight half-line divided into segments of unit length, which we will call positions. The positions are numbered by positive integers that start with 1 from the end of half-line, i. e. 1, 2, 3 and so on. The distance between the positions is the absolute difference between the respective numbers.

Laharl, Etna and Flonne occupy some positions on the half-line and they want to get to the position with the largest possible number. They are originally placed in different positions.

Each of the characters can perform each of the following actions **no more than once**:

1. Move a certain distance.
2. Grab another character and lift him above the head.
3. Throw the lifted character a certain distance.

Each character has a movement range parameter. They can only move to free positions, assuming that distance between those positions doesn't exceed the movement range. One character can lift another character if the distance between the two characters equals 1, and no one already holds that another character. We can assume that the lifted character moves to the same position as the person who has lifted him, and the position in which he stood before becomes free. A lifted character cannot perform any actions and the character that holds him cannot walk.

Also, each character has a throwing range parameter. It is the distance at which this character can throw the one lifted above his head. He can only throw a character to a free position, and only when there is a lifted character.

We accept the situation when one person grabs another one who in his turn has the third character in his hands. This forms a "column" of three characters. For example, Laharl can hold Etna while Etna holds Flonne. In this case, Etna and the Flonne cannot perform any actions, and Laharl can only throw Etna (together with Flonne) at some distance.

Laharl, Etna and Flonne perform actions in any order. They perform actions in turns, that is no two of them can do actions at the same time.

Determine the maximum number of position at least one of the characters can reach. That is, such maximal number x so that one of the characters can reach position x.

Input

The first line contains three integers: Laharl's position, his movement range and throwing range. The second and the third lines describe Etna's and Flonne's parameters correspondingly in the similar form. It is guaranteed that the three characters occupy distinct positions. All numbers in the input are between 1 and 10, inclusive.

Output

Print a single number — the maximum ordinal number of position which either Laharl, Etna or Flonne can reach.

Examples

input

9 3 3

4 3 1

2 3 3

output

15

Note

Let us explain how to reach position 15 in the sample.

Initially Laharl occupies position 9, Etna — position 4 and Flonne — position 2.

First Laharl moves to position 6.

Then Flonne moves to position 5 and grabs Etna.

Laharl grabs Flonne and throws to position 9.

Flonne throws Etna to position 12.

Etna moves to position 15.

Testcase:

6 5 10

9 7 10

10 10 10