A. Sum

1 second, 256 megabytes

You are given three integers a, b, and c. Determine if one of them is the sum of the other two.

Input

The first line contains a single integer t ($1 \leq t \leq 9261$) — the number of test cases.

The description of each test case consists of three integers a, b, c ($0 \leq a, b, c \leq 20$).

Output

For each test case, output "YES" if one of the numbers is the sum of the other two, and "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

input

```
7
1 4 3
2 5 8
9 11 20
0 0 0
20 20 20
4 12 3
15 7 8
```

output

```
YES
NO
YES
YES
NO
NO
YES
```

In the first test case, $1 + 3 = 4$.

In the second test case, none of the numbers is the sum of the other two.

In the third test case, $9 + 11 = 20$.

B. Division?

1 second, 256 megabytes

Codeforces separates its users into 4 divisions by their rating:

- For Division 1: $1900 \leq \mathrm{rating}$
- For Division 2: $1600 \leq \mathrm{rating} \leq 1899$
- For Division 3: $1400 \leq \mathrm{rating} \leq 1599$
- For Division 4: $\mathrm{rating} \leq 1399$

Given a rating, print in which division the rating belongs.

Input

The first line of the input contains an integer t ($1 \leq t \leq 10^4$) — the number of testcases.

The description of each test consists of one line containing one integer rating ($-5000 \leq \mathrm{rating} \leq 5000$).

Output

For each test case, output a single line containing the correct division in the format "Division X", where X is an integer between 1 and 4 representing the division for the corresponding rating.

input

```
7
-789
1299
1300
1399
1400
1679
2300
```

output

```
Division 4
Division 4
Division 4
Division 4
Division 3
Division 2
Division 1
```

For test cases $1 - 4$, the corresponding ratings are -789, 1299, 1300, 1399, so all of them are in division 4.

For the fifth test case, the corresponding rating is 1400, so it is in division 3.

For the sixth test case, the corresponding rating is 1679, so it is in division 2.

For the seventh test case, the corresponding rating is 2300, so it is in division 1.

C. Football

2 seconds, 256 megabytes

Petya loves football very much. One day, as he was watching a football match, he was writing the players' current positions on a piece of paper. To simplify the situation he depicted it as a string consisting of zeroes and ones. A zero corresponds to players of one team; a one corresponds to players of another team. If there are at least 7 players of some team standing one after another, then the situation is considered dangerous. For example, the situation 00100110111111101 is dangerous and 11110111011101 is not. You are given the current situation. Determine whether it is dangerous or not.

Input

The first input line contains a non-empty string consisting of characters "0" and "1", which represents players. The length of the string does not exceed 100 characters. There's at least one player from each team present on the field.

Output

Print "YES" if the situation is dangerous. Otherwise, print "NO".

input

```
001001
```

output

```
NO
```

input

```
1000000001
```

output

```
YES
```

D. Magic Numbers

2 seconds, 256 megabytes

A magic number is a number formed by concatenation of numbers 1, 14 and 144. We can use each of these numbers any number of times. Therefore 14144, 141414 and 1411 are magic numbers but 1444, 514 and 414 are not.

You're given a number. Determine if it is a magic number or not.

Input

The first line of input contains an integer n, $(1 \le n \le 10^9)$. This number doesn't contain leading zeros.

Output

Print "YES" if n is a magic number or print "NO" if it's not.

input

```
114114
```

output

```
YES
```

input

```
1111
```

output

```
YES
```

input

```
441231
```

output

```
NO
```

E. Stages

1 second, 256 megabytes

Natasha is going to fly to Mars. She needs to build a rocket, which consists of several stages in some order. Each of the stages is defined by a lowercase Latin letter. This way, the rocket can be described by the string — concatenation of letters, which correspond to the stages.

There are n stages available. The rocket must contain exactly k of them. Stages in the rocket should be ordered by their weight. So, after the stage with some letter can go only stage with a letter, which is at least two positions after in the alphabet (skipping one letter in between, or even more). For example, after letter 'c' can't go letters 'a', 'b', 'c' and 'd', but can go letters 'e', 'f', ..., 'z'.

For the rocket to fly as far as possible, its weight should be minimal. The weight of the rocket is equal to the sum of the weights of its stages. The weight of the stage is the number of its letter in the alphabet. For example, the stage 'a 'weighs one ton,' b 'weighs two tons, and' z' — 26 tons.

Build the rocket with the minimal weight or determine, that it is impossible to build a rocket at all. Each stage can be used at most once.

Input

The first line of input contains two integers — n and k ($1 \le k \le n \le 50$) – the number of available stages and the number of stages to use in the rocket.

The second line contains string s, which consists of exactly n lowercase Latin letters. Each letter defines a new stage, which can be used to build the rocket. Each stage can be used at most once.

Output

Print a single integer — the minimal total weight of the rocket or `-1`, if it is impossible to build the rocket at all.

input

```
5 3
xyabd
```

output

```
29
```

input

```
7 4
problem
```

output

```
34
```

input

```
2 2
ab
```

output

```
-1
```

input

```
12 1
abaabbaaabbb
```

output

```
1
```

In the first example, the following rockets satisfy the condition:

- "adx" (weight is $1 + 4 + 24 = 29$);
- "ady" (weight is $1 + 4 + 25 = 30$);
- "bdx" (weight is $2 + 4 + 24 = 30$);
- "bdy" (weight is $2 + 4 + 25 = 31$).

Rocket "adx" has the minimal weight, so the answer is 29.

In the second example, target rocket is "belo". Its weight is $2 + 5 + 12 + 15 = 34$.

In the third example, $n = k = 2$, so the rocket must have both stages: 'a' and 'b'. This rocket doesn't satisfy the condition, because these letters are adjacent in the alphabet. Answer is `-1`.

F. King Escape

1 second, 256 megabytes

Alice and Bob are playing chess on a huge chessboard with dimensions $n \times n$. Alice has a single piece left — a queen, located at (a_x, a_y), while Bob has only the king standing at (b_x, b_y). Alice thinks that as her queen is dominating the chessboard, victory is hers.

But Bob has made a devious plan to seize the victory for himself — he needs to march his king to (c_x, c_y) in order to claim the victory for himself. As Alice is distracted by her sense of superiority, **she no longer moves any pieces around, and it is only Bob who makes any turns.**

Bob will win if he can move his king from (b_x, b_y) to (c_x, c_y) **without ever getting in check**. Remember that a king can move to any of the 8 adjacent squares. A king is in check if it is on the same rank (i.e. row), file (i.e. column), or diagonal as the enemy queen.

Find whether Bob can win or not.

Input

The first line contains a single integer n ($3 \le n \le 1000$) — the dimensions of the chessboard.

The second line contains two integers a_x and a_y ($1 \le a_x, a_y \le n$) — the coordinates of Alice's queen.

The third line contains two integers b_x and b_y ($1 \le b_x, b_y \le n$) — the coordinates of Bob's king.

The fourth line contains two integers c_x and c_y ($1 \le c_x, c_y \le n$) — the coordinates of the location that Bob wants to get to.

It is guaranteed that Bob's king is currently not in check and the target location is not in check either.

Furthermore, the king is not located on the same square as the queen (i.e. $a_x \neq b_x$ or $a_y \neq b_y$), and the target does coincide neither with the queen's position (i.e. $c_x \neq a_x$ or $c_y \neq a_y$) nor with the king's position (i.e. $c_x \neq b_x$ or $c_y \neq b_y$).

Output

Print "YES" (without quotes) if Bob can get from (b_x, b_y) to (c_x, c_y) without ever getting in check, otherwise print "NO".

You can print each letter in any case (upper or lower).

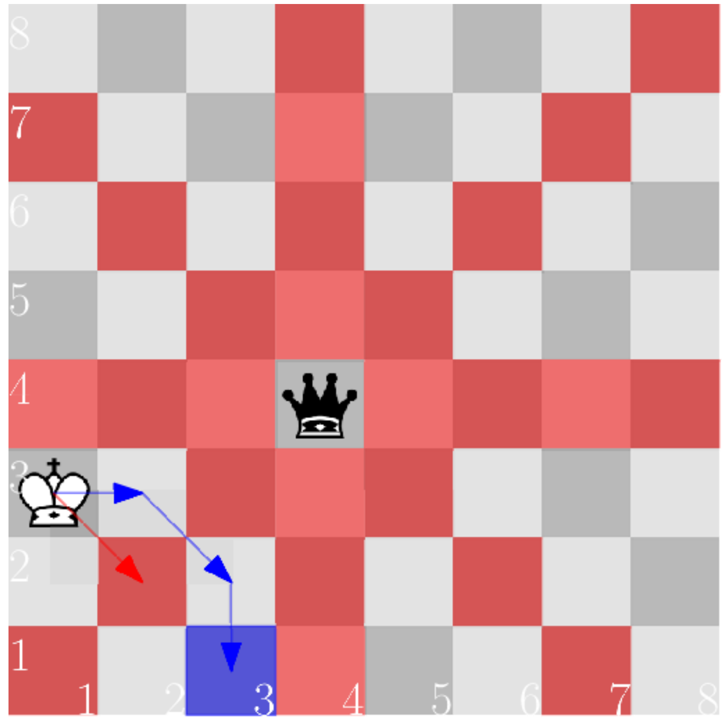
input
```
8
4 4
1 3
3 1
```
output
```
YES
```

input
```
8
4 4
2 3
1 6
```
output
```
NO
```

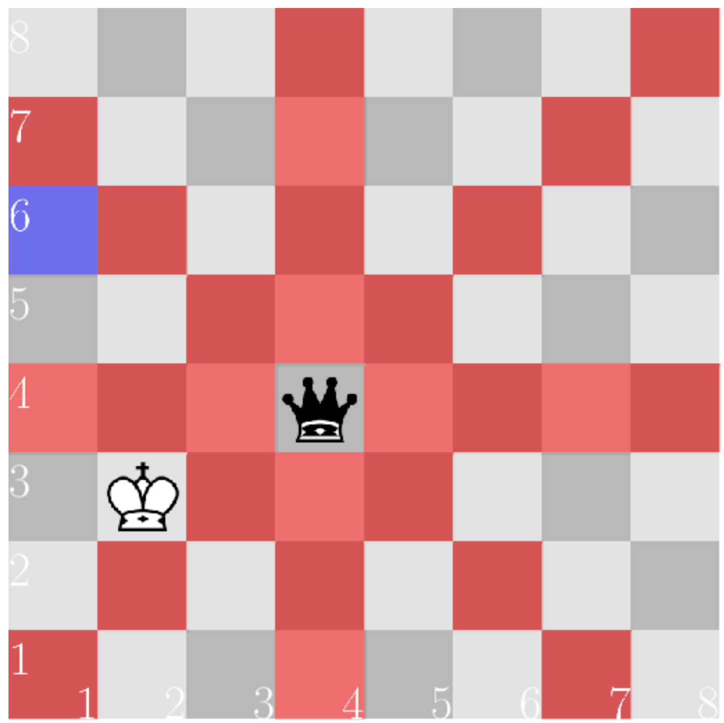
input
```
8
3 5
1 2
6 1
```
output
```
NO
```

In the diagrams below, the squares controlled by the black queen are marked red, and the target square is marked blue.

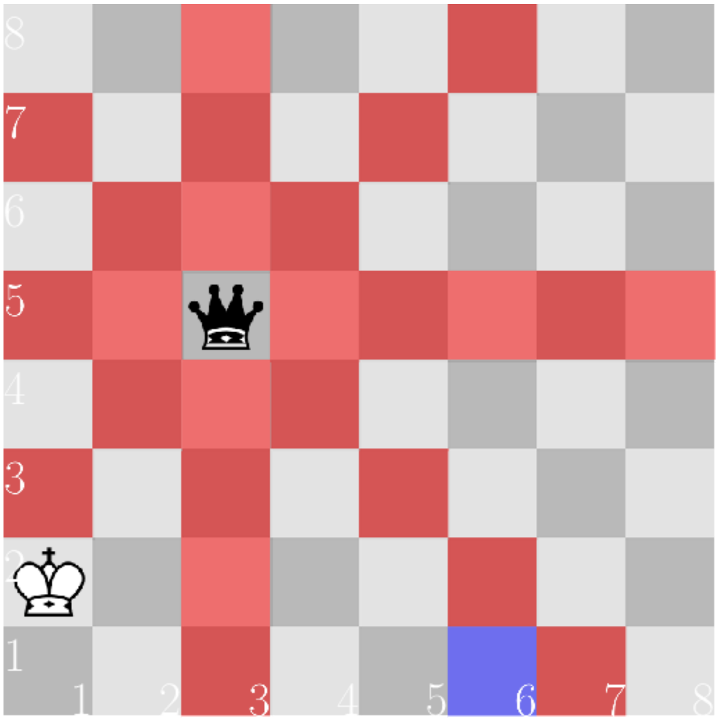
In the first case, the king can move, for instance, via the squares $(2, 3)$ and $(3, 2)$. Note that the direct route through $(2, 2)$ goes through check.



In the second case, the queen watches the fourth rank, and the king has no means of crossing it.

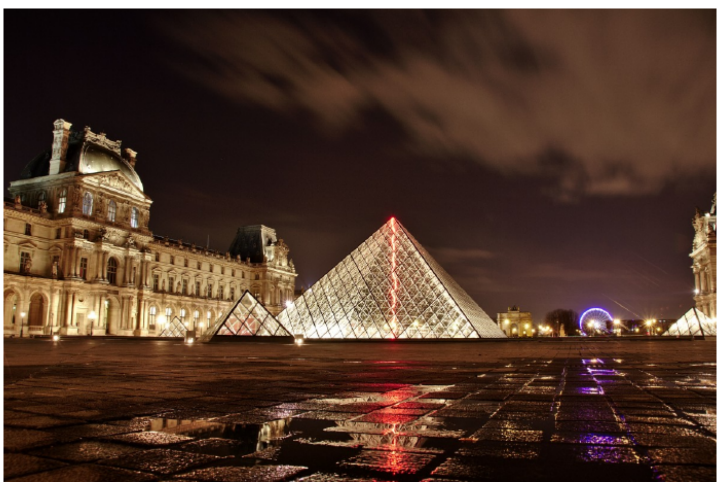


In the third case, the queen watches the third file.



G_Bonus. City of Lights

2.0 s, 256 megabytes



Paris has been called "ville lumière" (city of lights) since the 17th century. It earned this nickname in part because of the many city lights illuminating famous sites such as monuments, statues, churches, or fountains.

Those public lights in Paris are numbered from 1 to N and are all on by default. A group of hackers has gained the capability to toggle groups of lights. Every time the hackers use their program, they cause a number i (that they cannot control) to be sent to the system controlling the city lights. The lights numbered i, $2i$, $3i$, and so on (up to N) then change state instantly: lights that were on go off, and lights that were off go on.

During the night, the hackers use their programs k times. What is the greatest number of lights that are simultaneously off at the same time?

Input

The input comprises several lines, each consisting of a single integer:

- The first line contains the number N of lights. $1 \leq N \leq 1000000$.
- The second line contains the number k of uses hackers's program. $1 \leq k \leq 100$.
- The next k lines contain a number i sent to the system controlling the lights. $1 \leq i \leq N$.

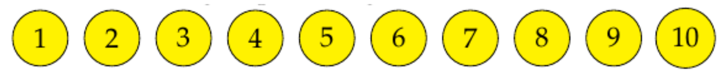
Output

The output should consist of a single line, whose content is an integer, the greatest number of lights that are simultaneously off at the same time.

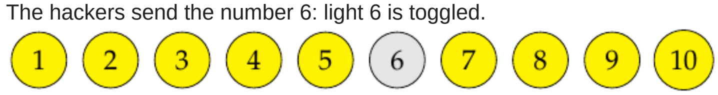
input
10 4 6 2 1 3
output
6

Sample Explanation:

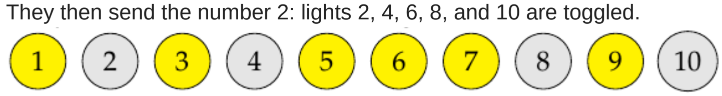
We start with a group of 10 lights which are all on.



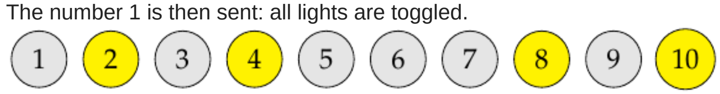
The hackers send the number 6: light 6 is toggled.



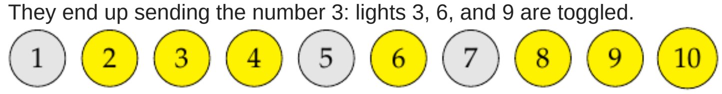
They then send the number 2: lights 2, 4, 6, 8, and 10 are toggled.



The number 1 is then sent: all lights are toggled.



They end up sending the number 3: lights 3, 6, and 9 are toggled.



The maximum number of lights off at the same time was 6.

H_Bonus. Gratitude

3 seconds, 256 megabytes

Ben heard about studies by Emmons and McCullough that suggest that intentionally practicing gratitude has a lasting effect on people's happiness. Since he wants to be happy too, he decided that at the end of each day he will think back over the past day and write down three things he is thankful for, one thing per line. At the end of N days in which he practiced this exercise, he was curious to know which things appear the most on his list. Help Ben get the K things he was grateful for most frequently.

Input

The input begins with one line containing two space-separated integers, N and K, in that order. Then follow $3N$ lines containing Ben's notes from N days. You may assume that the three lines that correspond to the same day contain no repetitions. That is, if you partition the input into N chunks of 3 consecutive lines, no chunk contains two identical lines.

Limits

- $1 \leq K \leq 3N \leq 100\,000$
- Each input line contains at most 50 (ASCII) characters.

Output

The output should represent the list of things that Ben is grateful for, ordered by frequency of appearance in Ben's list (with the most frequent item first). In case of two items with equal frequency, the most recent item should appear first. That is, in case of a tie in the number of appearances, the item whose last appearance is later in the input should appear earlier in the output. Finally, if there are more than K different items in Ben's list, your output should contain only the K first items (according to the required order).

input

```
2 2
Supportive parents
Being able to solve a hard problem
Good food
Fun game with friends
Good food
Being healthy
```

output

```
Good food
Being healthy
```

input

```
2 6
Supportive parents
Being able to solve a hard problem
Good food
Fun game with friends
Good food
Being healthy
```

output

```
Good food
Being healthy
Fun game with friends
Being able to solve a hard problem
Supportive parents
```

Sample Explanation 1

`Good food` is the only item that appears twice in Ben's list, so it should appear first in the output. All other items appear once in the input, but `Being healthy` takes precedence as it is the most recent.

Sample Explanation 1

Here there are only 5 different items that Ben is grateful for, so there are only 5 lines of output. In this list, `Good food` is first in the output since it appears twice in the input, and the other items are ordered by last appearance in Ben's list.

I_Bonus. Improving IT

3.0 s, 512 megabytes

Your best friend is part of the business team at the Global Center for Parallel Computing (GCPC). She is responsible for buying and selling the hardware that is powering the system that will be in use for the next n months. Currently, she is planning the CPU replacement cycle for a single CPU. To ensure that the system is always up-to-date, the CPU must be replaced at least every m months. Fortunately, she can sell the replaced CPU to lower the overall costs to operate the new system. However, storage capacity is pricey, and she has to accept the resale value the CPU has in the month it is replaced. That means, when a CPU that was used for j months is replaced in month i, you need to sell the current CPU for the value it has after j months of usage and buy a new CPU for the price of the ith month. She already compiled a list of CPU prices for the next n months including their resale value after 1 to m months. Note that you definitely need to buy a CPU in month 1 and you need to sell the last CPU in month $n+1$. How much money does the system cost at least over the n months?

Input

The input consists of:

- One line with two integers n and m ($1 \leq n, m \text{ and } n \cdot m \leq 5 \cdot 10^5$).
- n lines; the ith line has an integer c ($0 \leq c \leq 10^9$), the cost of a CPU in month i, followed by $\min(m, n-i+1)$ integers c_j ($0 \leq c_j \leq 10^9$), the money you earn by selling this CPU after $j > 0$ months.

Output

Output a single integer, the minimum total cost. Note that this number can be negative if reselling CPUs was profitable.

input

```
4 3
1000 900 800 900
700 600 500 400
1200 1200 1300
600 500
```

output

```
100
```

input

```
3 2
200 300 400
400 300 200
300 500
```

output

```
-400
```