A - ABC Preparation

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 100 points

Problem Statement

Takahashi has decided to hold some number of programming contests.

Holding one contest requires one 100-point problem, one 200-point problem, one 300-point problem, and one 400-point problem.

When he has A_1, A_2, A_3, and A_4 drafts of 100-, 200-, 300-, and 400-point problems, respectively, at most how many contests can he hold?

The same draft can be used only once.

Constraints

- $1 \leq A_i \leq 100 \, (1 \leq i \leq 4)$
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$A_1 \quad A_2 \quad A_3 \quad A_4$$

Output

Print an integer representing the maximum number of contests that can be held.

Sample Input 1

```
5 3 7 11
```

Sample Output 1

```
3
```

By using three drafts for each slot, he can hold three contests. He has just three drafts for 200-point problems, so he cannot hold four.

Sample Input 2

```
100 100 1 100
```

Sample Output 2

```
1
```

A contest cannot be held even if there is just one missing slot.

B - Smartphone Addiction

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 200 points

Problem Statement

The battery of Takahashi's smartphone has N mAh capacity. At time $0.5, 1.5, 2.5$, and so on (that is, at time $n + 0.5$ for every integer n), the battery charge decreases by 1 mAh.

Takahashi will leave his house with his phone fully charged at time 0, visit a cafe M times, and return home at time T.

He will stay at the i-th cafe from time A_i to time B_i. During this stay, he charges his phone, so the battery charge does not decrease. Instead, at time $n + 0.5$ for every integer n, it increases by 1. However, if it is already equal to the battery capacity, it does not increase nor decrease.

Determine whether he can return home without the battery charge dropping to 0 on the way.

Constraints

- $1 \leq N \leq 10^9$
- $1 \leq M \leq 1000$
- $1 \leq T \leq 10^9$
- $0 < A_1 < B_1 < A_2 < B_2 < A_3 < B_3 < \cdots < A_M < B_M < T$
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \quad T \\
A_1 \quad B_1 \\
A_2 \quad B_2 \\
A_3 \quad B_3 \\
\quad \vdots \\
A_M \quad B_M
\end{array}
$$

Output

If Takahashi can return home without the battery charge dropping to 0 on the way, print `Yes`; otherwise, print `No`.

Sample Input 1

```
10 2 20
9 11
13 17
```

Sample Output 1

```
Yes
```

The battery charge changes as follows:

- Time 0 (leaving home): 10 mAh
- Time 9 (the beginning of the stay at the first cafe): 1 mAh
- Time 11 (the end of the stay at the first cafe): 3 mAh (He charges his phone in a cafe.)
- Time 13 (the beginning of the stay at the second cafe): 1 mAh
- Time 17 (the end of the stay at the second cafe): 5 mAh
- Time 20 (getting home): 2 mAh

During this process, the battery charge never drops to 0, so we print `Yes`.

Sample Input 2

```
10 2 20
9 11
13 16
```

Sample Output 2

```
No
```

This case is the same as Sample Input/Output 1 until he starts his stay at the second cafe with 1 mAh charge.

When he ends his stay there at time 16, the battery charge is 4 mAh.

Then at time 19.5, it drops to 0, so we print `No`.

Sample Input 3

```
15 3 30
5 8
15 17
24 27
```

Sample Output 3

```
Yes
```

The battery charge drops to 1 mAh when he gets home, but it never drops to 0 on the way.

Sample Input 4

```
20 1 30
20 29
```

Sample Output 4

```
No
```

The battery charge drops to 0 at time 19.5.

Sample Input 5

```
20 1 30
1 10
```

Sample Output 5

```
No
```

Note that when the battery charge is equal to the battery capacity, staying at a cafe does not increase the battery charge.

C - Duodecim Ferra

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 300 points

Problem Statement

There is an iron bar of length L lying east-west. We will cut this bar at 11 positions to divide it into 12 bars. Here, each of the 12 resulting bars must have a positive integer length.

Find the number of ways to do this division. Two ways to do the division are considered different if and only if there is a position cut in only one of those ways.

Under the constraints of this problem, it can be proved that the answer is less than 2^{63}.

Constraints

- $12 \leq L \leq 200$
- L is an integer.

Input

Input is given from Standard Input in the following format:

```
L
```

Output

Print the number of ways to do the division.

Sample Input 1

```
12
```

Sample Output 1

```
1
```

There is only one way: to cut the bar into 12 bars of length 1 each.

Sample Input 2

```
13
```

Sample Output 2

```
12
```

Just one of the resulting bars will be of length 2. We have 12 options: one where the westmost bar is of length 2, one where the second bar from the west is of length 2, and so on.

Sample Input 3

```
17
```

Sample Output 3

```
4368
```

D - Stamp

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 400 points

Problem Statement

There are N squares arranged in a row from left to right. Let Square i be the i-th square from the left.
M of those squares, Square $A_1, A_2, A_3, \ldots, A_M$, are blue; the others are white. (M may be 0, in which case there is no blue square.)
You will choose a positive integer k just once and make a stamp with width k. In one use of a stamp with width k, you can choose k consecutive squares from the N squares and repaint them red, as long as those k squares do not contain a blue square.
At least how many uses of the stamp are needed to have no white square, with the optimal choice of k and the usage of the stamp?

Constraints

- $1 \leq N \leq 10^9$
- $0 \leq M \leq 2 \times 10^5$
- $1 \leq A_i \leq N$
- A_i are pairwise distinct.
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$N \quad M$$
$$A_1 \quad A_2 \quad A_3 \quad \ldots \quad A_M$$

Output

Print the minimum number of uses of the stamp needed to have no white square.

Sample Input 1

```
5 2
1 3
```

Sample Output 1

```
3
```

If we choose $k = 1$ and repaint the three white squares one at a time, three uses are enough, which is optimal.
Choosing $k = 2$ or greater would make it impossible to repaint Square 2, because of the restriction that does not allow the k squares to contain a blue square.

Sample Input 2

```
13 3
13 3 9
```

Sample Output 2

```
6
```

One optimal strategy is choosing $k = 2$ and using the stamp as follows:

- Repaint Squares $1, 2$ red.
- Repaint Squares $4, 5$ red.
- Repaint Squares $5, 6$ red.
- Repaint Squares $7, 8$ red.
- Repaint Squares $10, 11$ red.
- Repaint Squares $11, 12$ red.

Note that, although the k consecutive squares chosen when using the stamp cannot contain blue squares, they can contain already red squares.

Sample Input 3

```
5 5
5 2 1 4 3
```

Sample Output 3

```
0
```

If there is no white square from the beginning, we do not have to use the stamp at all.

Sample Input 4

```
1 0
```

Sample Output 4

```
1
```

M may be 0.

E - Sequence Matching

Time Limit: 2 sec / Memory Limit: 1024 MB

Score : 500 points

Problem Statement

We have an integer sequence A of length N and an integer sequence B of length M.
Takahashi will make a new sequence A' by removing some elements (possibly zero or all) from A and concatenating the remaining elements.
Similarly, he will make another new sequence B' by removing some elements (possibly zero or all) from B and concatenating the remaining elements.
Here, he will remove elements so that $|A'| = |B'|$. ($|s|$ denotes the length of s for a sequence s.)
Let x be the total number of elements removed from A and B, and y be the number of integers i such that $1 \leq i \leq |A'|$ and $A'_i \neq B'_i$. Print the minimium possible value of $x + y$.

Constraints

- $1 \leq N, M \leq 1000$
- $1 \leq A_i, B_i \leq 10^9$
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
A_1 \quad A_2 \quad A_3 \quad \ldots \quad A_N \\
B_1 \quad B_2 \quad B_3 \quad \ldots \quad B_M
\end{array}
$$

Output

Print the minimum possible value of $x + y$.

Sample Input 1

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

Sample Output 1

```
2
```

If we make A' by removing A_4 from A, and B' by removing nothing from B, x will be 1.
Here, there is just one integer i such that $1 \leq i \leq |A'|$ and $A'_i \neq B'_i$: $i = 2$, so y will be 1, and $x + y$ will be 2, which is the minimum possible value.

Sample Input 2

```
4 6
1 3 2 4
1 5 2 6 4 3
```

Sample Output 2

```
3
```

If we remove nothing from A and remove B_4, B_6 from B, we have $x = 2, y = 1$, and $x + y = 3$, which is the minimum possible value.

Sample Input 3

```
5 5
1 1 1 1 1
2 2 2 2 2
```

Sample Output 3

```
5
```

It is allowed to remove nothing from both A and B.

F - Range Xor Query

Time Limit: 3 sec / Memory Limit: 1024 MB

Score : 600 points

Problem Statement

We have an integer sequence A of length N.

You will process Q queries on this sequence. In the i-th query, given values T_i, X_i, and Y_i, do the following:

- If $T_i = 1$, replace A_{X_i} with $A_{X_i} \oplus Y_i$.
- If $T_i = 2$, print $A_{X_i} \oplus A_{X_i+1} \oplus A_{X_i+2} \oplus \cdots \oplus A_{Y_i}$.

Here, $a \oplus b$ denotes the bitwise XOR of a and b.

▶ What is bitwise XOR?

Constraints

- $1 \leq N \leq 300000$
- $1 \leq Q \leq 300000$
- $0 \leq A_i < 2^{30}$
- T_i is 1 or 2.
- If $T_i = 1$, then $1 \leq X_i \leq N$ and $0 \leq Y_i < 2^{30}$.
- If $T_i = 2$, then $1 \leq X_i \leq Y_i \leq N$.
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{lll}
N & Q & \\
A_1 & A_2 & A_3 \ \ldots \ A_N \\
T_1 & X_1 & Y_1 \\
T_2 & X_2 & Y_2 \\
T_3 & X_3 & Y_3 \\
 & \vdots & \\
T_Q & X_Q & Y_Q
\end{array}
$$

Output

For each query with $T_i = 2$ in the order received, print the response in its own line.

Sample Input 1

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

Sample Output 1

```
0
1
2
```

In the first query, we print $1 \oplus 2 \oplus 3 = 0$.

In the second query, we print $2 \oplus 3 = 1$.

In the third query, we replace A_2 with $2 \oplus 3 = 1$.

In the fourth query, we print $1 \oplus 3 = 2$.

Sample Input 2

```
10 10
0 5 3 4 7 0 0 0 1 0
1 10 7
2 8 9
2 3 6
2 1 6
2 1 10
1 9 4
1 6 1
1 6 3
1 1 7
2 3 5
```

Sample Output 2

```
1
0
5
3
0
```