A. Perfect Permutation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a positive integer n.

The weight of a permutation $p_1, p_2, \ldots, p_n$ is the number of indices $1 \le i \le n$ such that i divides p_i. Find a permutation $p_1, p_2, \ldots, p_n$ with the minimum possible weight (among all permutations of length n).

A permutation is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (2 appears twice in the array) and $[1,3,4]$ is also not a permutation ($n=3$ but there is 4 in the array).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The only line of each test case contains a single integer n ($1 \le n \le 10^5$) — the length of permutation.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, print a line containing n integers $p_1, p_2, \ldots, p_n$ so that the permutation p has the minimum possible weight.

If there are several possible answers, you can print any of them.

Example

input

```
2
1
4
```

output

```
1
2 1 4 3
```

Note

In the first test case, the only valid permutation is $p=[1]$. Its weight is 1.

In the second test case, one possible answer is the permutation $p=[2,1,4,3]$. One can check that 1 divides p_1 and i does not divide p_i for $i=2,3,4$, so the weight of this permutation is 1. It is impossible to find a permutation of length 4 with a strictly smaller weight.

B. Party

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A club plans to hold a party and will invite some of its n members. The n members are identified by the numbers $1, 2, \ldots, n$. If member i is not invited, the party will gain an unhappiness value of a_i.

There are m pairs of friends among the n members. As per tradition, if both people from a friend pair are invited, they will share a cake at the party. The total number of cakes eaten will be equal to the number of pairs of friends such that both members have been invited.

However, the club's oven can only cook two cakes at a time. So, the club demands that the total number of cakes eaten is an even number.

What is the minimum possible total unhappiness value of the party, respecting the constraint that the total number of cakes eaten is even?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n \le 10^5$, $0 \le m \le \min(10^5, \frac{n(n-1)}{2})$) — the number of club members and pairs of friends.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 10^4$) — the unhappiness value array.

Each of the next m lines contains two integers x and y ($1 \le x, y \le n$, $x \ne y$) indicating that x and y are friends. Each unordered pair (x, y) appears at most once in each test case.

It is guaranteed that both the sum of n and the sum of m over all test cases do not exceed 10^5.

Output

For each test case, print a line containing a single integer – the minimum possible unhappiness value of a valid party.

Example

input

```
4
1 0
1
3 1
2 1 3
1 3
5 5
1 2 3 4 5
1 2
1 3
1 4
1 5
2 3
5 5
1 1 1 1 1
1 2
2 3
3 4
4 5
5 1
```

output

```
0
2
3
2
```

Note

In the first test case, all members can be invited. So the unhappiness value is 0.

In the second test case, the following options are possible:

- invite 1 and 2 (0 cakes eaten, unhappiness value equal to 3);
- invite 2 and 3 (0 cakes eaten, unhappiness value equal to 2);
- invite only 1 (0 cakes eaten, unhappiness value equal to 4);
- invite only 2 (0 cakes eaten, unhappiness value equal to 5);
- invite only 3 (0 cakes eaten, unhappiness value equal to 3);
- invite nobody (0 cakes eaten, unhappiness value equal to 6).

The minimum unhappiness value is achieved by inviting 2 and 3.
In the third test case, inviting members $3, 4, 5$ generates a valid party with the minimum possible unhappiness value.

C. Color the Picture

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A picture can be represented as an $n \times m$ grid (n rows and m columns) so that each of the $n \cdot m$ cells is colored with one color. You have k pigments of different colors. You have a limited amount of each pigment, more precisely you can color at most a_i cells with the i-th pigment.

A picture is considered beautiful if each cell has **at least** 3 **toroidal** neighbors with the same color as itself.

Two cells are considered **toroidal** neighbors if they **toroidally** share an edge. In other words, for some integers $1 \le x_1, x_2 \le n$ and $1 \le y_1, y_2 \le m$, the cell in the x_1-th row and y_1-th column is a toroidal neighbor of the cell in the x_2-th row and y_2-th column if one of following two conditions holds:

- $x_1 - x_2 \equiv \pm 1 \pmod{n}$ and $y_1 = y_2$, or
- $y_1 - y_2 \equiv \pm 1 \pmod{m}$ and $x_1 = x_2$.

Notice that each cell has exactly 4 toroidal neighbors. For example, if $n = 3$ and $m = 4$, the toroidal neighbors of the cell $(1, 2)$ (the cell on the first row and second column) are: $(3, 2)$, $(2, 2)$, $(1, 3)$, $(1, 1)$. They are shown in gray on the image below:

The gray cells show toroidal neighbors of $(1,2)$.

Is it possible to color all cells with the pigments provided and create a beautiful picture?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains three integers n, m, and k ($3 \le n, m \le 10^9$, $1 \le k \le 10^5$) — the number of rows and columns of the picture and the number of pigments.

The next line contains k integers $a_1, a_2, \dots, a_k$ ($1 \le a_i \le 10^9$) — a_i is the maximum number of cells that can be colored with the i-th pigment.

It is guaranteed that the sum of k over all test cases does not exceed 10^5.

Output

For each test case, print "Yes" (without quotes) if it is possible to color a beautiful picture. Otherwise, print "No" (without quotes).

Example

input

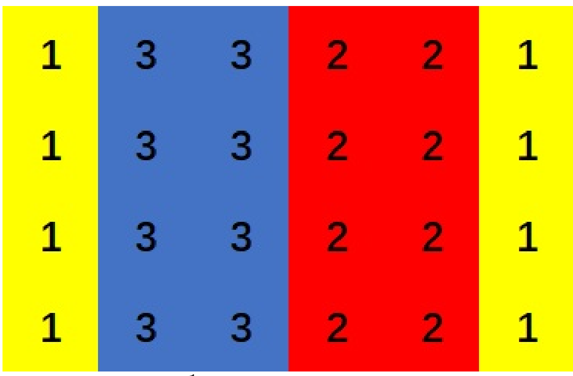
```
6
4 6 3
12 9 8
3 3 2
8 8
3 3 2
9 5
4 5 2
10 11
5 4 2
9 11
10 10 3
11 45 14
```

output

```
Yes
No
Yes
Yes
No
No
```

Note

In the first test case, one possible solution is as follows:



In the third test case, we can color all cells with pigment 1.

D. Rain

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are the owner of a harvesting field which can be modeled as an infinite line, whose positions are identified by integers.

It will rain for the next n days. On the i-th day, the rain will be centered at position x_i and it will have intensity p_i. Due to these rains, some rainfall will accumulate; let a_j be the amount of rainfall accumulated at integer position j. Initially a_j is 0, and it will increase by $\max(0, p_i - |x_i - j|)$ after the i-th day's rain.

A flood will hit your field if, at any moment, there is a position j with accumulated rainfall $a_j > m$.

You can use a magical spell to erase **exactly one** day's rain, i.e., setting $p_i = 0$. For each i from 1 to n, check whether in case of erasing the i-th day's rain there is no flood.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n \le 2 \cdot 10^5$, $1 \le m \le 10^9$) — the number of rainy days and the maximal accumulated rainfall with no flood occurring.

Then n lines follow. The i-th of these lines contains two integers x_i and p_i ($1 \le x_i, p_i \le 10^9$) — the position and intensity of the i-th day's rain.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a binary string s length of n. The i-th character of s is 1 if after erasing the i-th day's rain there is **no** flood, while it is `0`, if after erasing the i-th day's rain the flood still happens.

Example

input

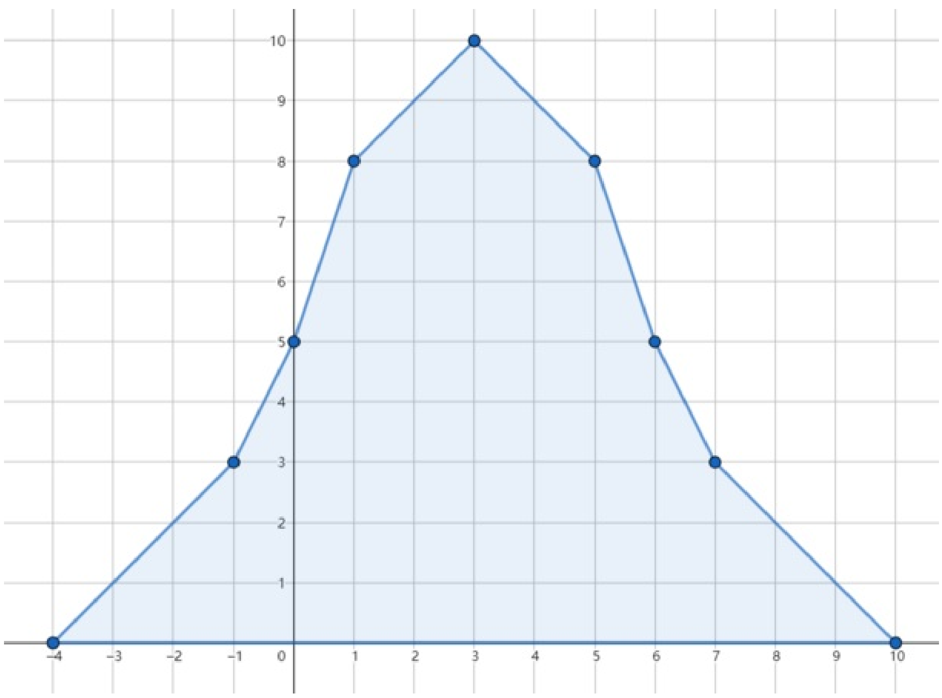
```
4
3 6
1 5
5 5
3 4
2 3
1 3
5 2
2 5
1 6
10 6
6 12
4 5
1 6
12 5
5 5
9 7
8 3
```

output

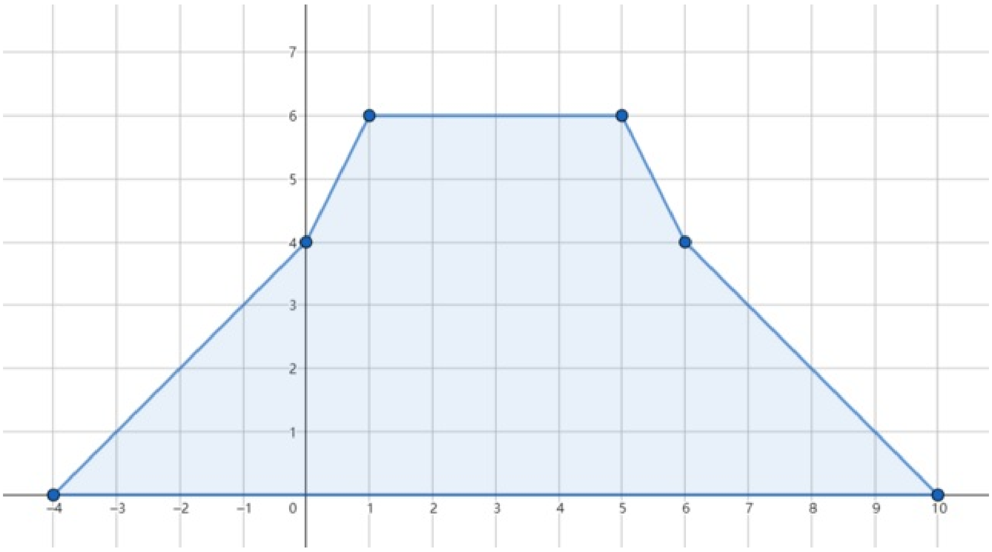
```
001
11
00
100110
```

Note

In the first test case, if we do not use the spell, the accumulated rainfall distribution will be like this:



If we erase the third day's rain, the flood is avoided and the accumulated rainfall distribution looks like this:



In the second test case, since initially the flood will not happen, we can erase any day's rain.

In the third test case, there is no way to avoid the flood.

E. XOR Triangle

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a positive integer n. Since n may be very large, you are given its binary representation.

You should compute the number of triples (a, b, c) with $0 \leq a, b, c \leq n$ such that $a \oplus b$, $b \oplus c$, and $a \oplus c$ are the sides of a non-degenerate triangle.

Here, $\oplus$ denotes the bitwise XOR operation.

You should output the answer modulo $998\,244\,353$.

Three positive values x, y, and z are the sides of a non-degenerate triangle if and only if $x + y > z$, $x + z > y$, and $y + z > x$.

Input

The first and only line contains the binary representation of an integer n ($0 < n < 2^{200\,000}$) without leading zeros.

For example, the string `10` is the binary representation of the number 2, while the string `1010` represents the number 10.

Output

Print one integer — the number of triples (a, b, c) satisfying the conditions described in the statement modulo $998\,244\,353$.

Examples

input
```
101
```
output
```
12
```

input
```
1110
```
output
```
780
```

input
```
11011111101010010
```
output
```
141427753
```

Note

In the first test case, $101_2 = 5$.

- The triple $(a, b, c) = (0, 3, 5)$ is valid because $(a \oplus b, b \oplus c, c \oplus a) = (3, 6, 5)$ are the sides of a non-degenerate triangle.
- The triple $(a, b, c) = (1, 2, 4)$ is valid because $(a \oplus b, b \oplus c, c \oplus a) = (3, 6, 5)$ are the sides of a non-degenerate triangle.

The 6 permutations of each of these two triples are all the valid triples, thus the answer is 12.

In the third test case, $11\,011\,111\,101\,010\,010_2 = 114\,514$. The full answer (before taking the modulo) is $1\,466\,408\,118\,808\,164$.
