A. Acacius and String

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Acacius is studying strings theory. Today he came with the following problem.

You are given a string s of length n consisting of lowercase English letters and question marks. It is possible to replace question marks with lowercase English letters in such a way that a string "abacaba" occurs as a substring in a resulting string exactly once?

Each question mark should be replaced with **exactly one** lowercase English letter. For example, string "a?b?c" can be transformed into strings "aabbc" and "azbzc", but can't be transformed into strings "aabc", "a?bbc" and "babbc".

Occurrence of a string t of length m in the string s of length n as a substring is a index i ($1 \leq i \leq n - m + 1$) such that string $s[i..i+m-1]$ consisting of m consecutive symbols of s starting from i-th equals to string t. For example string "ababa" has two occurrences of a string "aba" as a substring with $i = 1$ and $i = 3$, but there are no occurrences of a string "aba" in the string "acba" as a substring.

Please help Acacius to check if it is possible to replace all question marks with lowercase English letters in such a way that a string "abacaba" occurs as a substring in a resulting string **exactly once**.

Input

First line of input contains an integer T ($1 \leq T \leq 5000$), number of test cases. T pairs of lines with test case descriptions follow.

The first line of a test case description contains a single integer n ($7 \leq n \leq 50$), length of a string s.

The second line of a test case description contains string s of length n consisting of lowercase English letters and question marks.

Output

For each test case output an answer for it.

In case if there is no way to replace question marks in string s with a lowercase English letters in such a way that there is exactly one occurrence of a string "abacaba" in the resulting string as a substring output "No".

Otherwise output "Yes" and in the next line output a resulting string consisting of n lowercase English letters. If there are multiple possible strings, output any.

You may print every letter in "Yes" and "No" in any case you want (so, for example, the strings yEs, yes, Yes, and YES will all be recognized as positive answer).

Example

input

```
6
7
abacaba
7
???????
11
aba?abacaba
11
abacaba?aba
15
asdf???f???qwer
11
abacabacaba
```

output

```
Yes
abacaba
Yes
abacaba
Yes
abadabacaba
Yes
abacabadaba
No
No
```

Note

In first example there is exactly one occurrence of a string "abacaba" in the string "abacaba" as a substring.

In second example seven question marks can be replaced with any seven lowercase English letters and with "abacaba" in particular.

In sixth example there are two occurrences of a string "abacaba" as a substring.

B. Dubious Cyrpto

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Pasha loves to send strictly positive integers to his friends. Pasha cares about security, therefore when he wants to send an integer n, he encrypts it in the following way: he picks three integers a, b and c such that $l \leq a, b, c \leq r$, and then he computes the encrypted value $m = n \cdot a + b - c$.

Unfortunately, an adversary intercepted the values l, r and m. Is it possible to recover the original values of a, b and c from this information? More formally, you are asked to find *any* values of a, b and c such that

- a, b and c are integers,
- $l \leq a, b, c \leq r$,
- there exists a strictly positive integer n, such that $n \cdot a + b - c = m$.

Input

The first line contains the only integer t ($1 \leq t \leq 20$) — the number of test cases. The following t lines describe one test case each.

Each test case consists of three integers l, r and m ($1 \leq l \leq r \leq 500\,000$, $1 \leq m \leq 10^{10}$). The numbers are such that the answer to the problem exists.

Output

For each test case output three integers a, b and c such that, $l \leq a, b, c \leq r$ and there exists a strictly positive integer n such that $n \cdot a + b - c = m$. It is guaranteed that there is at least one possible solution, and you can output any possible combination if there are multiple solutions.

Example

input

```
2
4 6 13
2 3 1
```

output

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

Note

In the first example $n = 3$ is possible, then $n \cdot 4 + 6 - 5 = 13 = m$. Other possible solutions include: $a = 4$, $b = 5$, $c = 4$ (when $n = 3$); $a = 5$, $b = 4$, $c = 6$ (when $n = 3$); $a = 6$, $b = 6$, $c = 5$ (when $n = 2$); $a = 6$, $b = 5$, $c = 4$ (when $n = 2$).

In the second example the only possible case is $n = 1$: in this case $n \cdot 2 + 2 - 3 = 1 = m$. Note that, $n = 0$ is not possible, since in that case n is not a strictly positive integer.

C. Choosing flowers

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Vladimir would like to prepare a present for his wife: they have an anniversary! He decided to buy her **exactly** n flowers.

Vladimir went to a flower shop, and he was amazed to see that there are m types of flowers being sold there, and there is unlimited supply of flowers of each type. Vladimir wants to choose flowers to maximize the happiness of his wife. He knows that after receiving the first flower of the i-th type happiness of his wife increases by a_i and after receiving each consecutive flower of this type her happiness increases by b_i. That is, if among the chosen flowers there are $x_i > 0$ flowers of type i, his wife gets $a_i + (x_i - 1) \cdot b_i$ additional happiness (and if there are no flowers of type i, she gets nothing for this particular type).

Please help Vladimir to choose exactly n flowers to maximize the total happiness of his wife.

Input

The first line contains the only integer t ($1 \leq t \leq 10\,000$), the number of test cases. It is followed by t descriptions of the test cases.

Each test case description starts with two integers n and m ($1 \leq n \leq 10^9$, $1 \leq m \leq 100\,000$), the number of flowers Vladimir needs to choose and the number of types of available flowers.

The following m lines describe the types of flowers: each line contains integers a_i and b_i ($0 \leq a_i, b_i \leq 10^9$) for i-th available type of flowers.

The test cases are separated by a blank line. It is guaranteed that the sum of values m among all test cases does not exceed $100\,000$.

Output

For each test case output a single integer: the maximum total happiness of Vladimir's wife after choosing exactly n flowers optimally.

Example

input

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

5 3
5 2
4 2
3 1
```

output

```
14
16
```

Note

In the first example case Vladimir can pick 1 flower of the first type and 3 flowers of the second type, in this case the total happiness equals $5 + (1 + 2 \cdot 4) = 14$.

In the second example Vladimir can pick 2 flowers of the first type, 2 flowers of the second type, and 1 flower of the third type, in this case the total happiness equals $(5 + 1 \cdot 2) + (4 + 1 \cdot 2) + 3 = 16$.

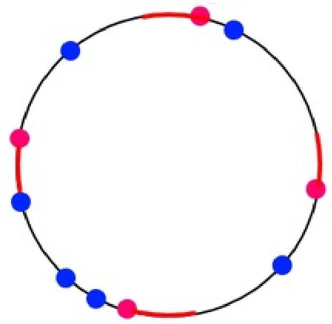
D. New Passenger Trams

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

There are many freight trains departing from Kirnes planet every day. One day on that planet consists of h hours, and each hour consists of m minutes, where m is an even number. Currently, there are n freight trains, and they depart every day at the same time: i-th train departs at h_i hours and m_i minutes.

The government decided to add passenger trams as well: they plan to add a regular tram service with half-hour intervals. It means that the first tram of the day must depart at 0 hours and t minutes, where $0 \le t < \frac{m}{2}$, the second tram departs $\frac{m}{2}$ minutes after the first one and so on. This schedule allows exactly two passenger trams per hour, which is a great improvement.

To allow passengers to board the tram safely, the tram must arrive k minutes before. During the time when passengers are boarding the tram, no freight train can depart from the planet. However, freight trains are allowed to depart at the very moment when the boarding starts, as well as at the moment when the passenger tram departs. Note that, if the first passenger tram departs at 0 hours and t minutes, where $t < k$, then the freight trains can not depart during the last $k - t$ minutes of the day.



A schematic picture of the correct way to run passenger trams. Here $h = 2$ (therefore, the number of passenger trams is $2h = 4$), the number of freight trains is $n = 6$. The passenger trams are marked in red (note that the spaces between them are the same). The freight trains are marked in blue. Time segments of length k before each passenger tram are highlighted in red. Note that there are no freight trains inside these segments.

Unfortunately, it might not be possible to satisfy the requirements of the government without canceling some of the freight trains. Please help the government find the optimal value of t to minimize the number of canceled freight trains in case all passenger trams depart according to schedule.

Input

The first line of input contains four integers n, h, m, k ($1 \le n \le 100\,000$, $1 \le h \le 10^9$, $2 \le m \le 10^9$, m is even, $1 \le k \le \frac{m}{2}$) — the number of freight trains per day, the number of hours and minutes on the planet, and the boarding time for each passenger tram.

n lines follow, each contains two integers h_i and m_i ($0 \le h_i < h$, $0 \le m_i < m$) — the time when i-th freight train departs. It is guaranteed that no freight trains depart at the same time.

Output

The first line of output should contain two integers: the minimum number of trains that need to be canceled, and the optimal starting time t. Second line of output should contain freight trains that need to be canceled.

Examples

input

```
2 24 60 15
16 0
17 15
```

output

```
0 0
```

input

```
2 24 60 16
16 0
17 15
```

output

```
1 0
2
```

Note

In the first test case of the example the first tram can depart at 0 hours and 0 minutes. Then the freight train at 16 hours and 0 minutes can depart at the same time as the passenger tram, and the freight train at 17 hours and 15 minutes can depart at the same time as the boarding starts for the upcoming passenger tram.

In the second test case of the example it is not possible to design the passenger tram schedule without cancelling any of the freight trains: if $t \in [1, 15]$, then the freight train at 16 hours and 0 minutes is not able to depart (since boarding time is 16 minutes). If $t = 0$ or $t \in [16, 29]$, then the freight train departing at 17 hours 15 minutes is not able to depart. However, if the second freight train is canceled, one can choose $t = 0$. Another possible option is to cancel the first train and choose $t = 13$.

E. Inverse Genealogy

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ivan is fond of genealogy. Currently he is studying a particular genealogical structure, which consists of some people. In this structure every person has either both parents specified, or none. Additionally, each person has exactly one child, except for one special person, who does not have any children. The people in this structure are conveniently numbered from 1 to n, and s_i denotes the child of the person i (and $s_i = 0$ for exactly one person who does not have any children).

We say that a is an ancestor of b if either $a = b$, or a has a child, who is an ancestor of b. That is a is an ancestor for a, s_a, s_{s_a}, etc.

We say that person i is *imbalanced* in case this person has both parents specified, and the total number of ancestors of one of the parents is at least double the other.

Ivan counted the number of imbalanced people in the structure, and got k people in total. However, he is not sure whether he computed it correctly, and would like to check if there is at least one construction with n people that have k imbalanced people in total. Please help him to find one such construction, or determine if it does not exist.

Input

The input contains two integers n and k ($1 \le n \le 100\,000$, $0 \le k \le n$), the total number of people and the number of imbalanced people.

Output

If there are no constructions with n people and k imbalanced people, output NO.

Otherwise output YES on the first line, and then n integers $s_1, s_2, \ldots, s_n$ ($0 \le s_i \le n$), which describes the construction and specify the child of each node (or 0, if the person does not have any children).

Examples

input

```
3 0
```

output

```
YES
0 1 1
```

input

```
5 1
```

output

```
YES
0 1 1 3 3
```

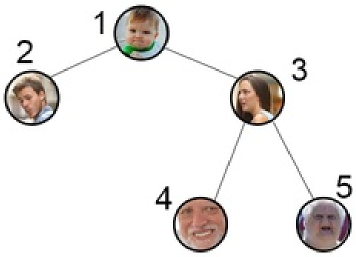
input
3 2

output
NO

Note

In the first example case one can have a construction with 3 people, where 1 person has 2 parents.

In the second example case one can use the following construction:



Only person 1 is imbalanced, because one of their parents has 1 ancestor in total, and the other parent has 3 ancestors.

F1. Chess Strikes Back (easy version)

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Note that the difference between easy and hard versions is that in hard version unavailable cells can become available again and in easy version can't. You can make hacks only if all versions are solved.

Ildar and Ivan are tired of chess, but they really like the chessboard, so they invented a new game. The field is a chessboard $2n \times 2m$: it has $2n$ rows, $2m$ columns, and the cell in row i and column j is colored white if $i+j$ is even, and is colored black otherwise.

The game proceeds as follows: Ildar marks some of the **white** cells of the chessboard as *unavailable*, and asks Ivan to place $n \times m$ kings on the remaining **white** cells in such way, so that there are no kings attacking each other. A king can attack another king if they are located in the adjacent cells, sharing an edge or a corner.

Ildar would like to explore different combinations of cells. Initially all cells are marked as available, and then he has q queries. In each query he marks a cell as unavailable. After each query he would like to know whether it is possible to place the kings on the available cells in a desired way. Please help him!

Input

The first line of input contains three integers n, m, q ($1 \leq n, m, q \leq 200\,000$) — the size of the board and the number of queries.

q lines follow, each of them contains a description of a query: two integers i and j, denoting a white cell (i, j) on the board ($1 \leq i \leq 2n$, $1 \leq j \leq 2m$, $i+j$ is even) that becomes unavailable. It's guaranteed, that each cell (i, j) appears in input at most once.

Output

Output q lines, i-th line should contain answer for a board after i queries of Ildar. This line should contain "YES" if it is possible to place the kings on the available cells in the desired way, or "NO" otherwise.

Examples

input
1 3 3 1 1 1 5 2 4

output
YES YES NO

input
3 2 7 4 2 6 4 1 3 2 2 2 4 4 4 3 1

output

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

Note

In the first example case after the second query only cells $(1, 1)$ and $(1, 5)$ are unavailable. Then Ivan can place three kings on cells $(2, 2)$, $(2, 4)$ and $(2, 6)$.

After the third query three cells $(1, 1)$, $(1, 5)$ and $(2, 4)$ are unavailable, so there remain only 3 available cells: $(2, 2)$, $(1, 3)$ and $(2, 6)$. Ivan can not put 3 kings on those cells, because kings on cells $(2, 2)$ and $(1, 3)$ attack each other, since these cells share a corner.

F2. Chess Strikes Back (hard version)

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Note that the difference between easy and hard versions is that in hard version unavailable cells can become available again and in easy version can't. You can make hacks only if all versions are solved.

Ildar and Ivan are tired of chess, but they really like the chessboard, so they invented a new game. The field is a chessboard $2n \times 2m$: it has $2n$ rows, $2m$ columns, and the cell in row i and column j is colored white if $i + j$ is even, and is colored black otherwise.

The game proceeds as follows: Ildar marks some of the **white** cells of the chessboard as *unavailable*, and asks Ivan to place $n \times m$ kings on the remaining **white** cells in such way, so that there are no kings attacking each other. A king can attack another king if they are located in the adjacent cells, sharing an edge or a corner.

Ildar would like to explore different combinations of cells. Initially all cells are marked as available, and then he has q queries. In each query he either marks a cell as unavailable, or marks the previously unavailable cell as available. After each query he would like to know whether it is possible to place the kings on the available cells in a desired way. Please help him!

Input

The first line of input contains three integers n, m, q ($1 \le n, m, q \le 200\,000$) — the size of the board and the number of queries.

q lines follow, each of them contains a description of a query: two integers i and j, denoting a white cell on the board ($1 \le i \le 2n$, $1 \le j \le 2m$, $i + j$ is even). If the cell (i, j) was available before the query, then it becomes unavailable. Otherwise, if the cell was unavailable, it becomes available.

Output

Output q lines, i-th line should contain answer for a board after i queries of Ildar. This line should contain "YES" if it is possible to place the kings on the available cells in the desired way, or "NO" otherwise.

Examples

input

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

output

```
YES
YES
NO
```

input

```
3 2 10
4 2
6 4
1 3
4 2
6 4
2 2
2 4
1 3
4 4
3 1
```

output

```
YES
YES
NO
NO
YES
```

```
YES
NO
YES
YES
NO
```

Note

In the first example case after the second query only cells $(1, 1)$ and $(1, 5)$ are unavailable. Then Ivan can place three kings on cells $(2, 2)$, $(2, 4)$ and $(2, 6)$.

After the third query three cells $(1, 1)$, $(1, 5)$ and $(2, 4)$ are unavailable, so there remain only 3 available cells: $(2, 2)$, $(1, 3)$ and $(2, 6)$. Ivan can not put 3 kings on those cells, because kings on cells $(2, 2)$ and $(1, 3)$ attack each other, since these cells share a corner.