Codeforces Round #449 (Div. 1)

A. Nephren gives a riddle

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

What are you doing at the end of the world? Are you busy? Will you save us?

Nephren is playing a game with little leprechauns.

She gives them an infinite array of strings, $f_{0\ldots\infty}$.

f_0 is "`What are you doing at the end of the world? Are you busy? Will you save us?`".

She wants to let more people know about it, so she defines $f_i =$ "`What are you doing while sending "`f_{i-1}"`? Are you busy? Will you send "`f_{i-1}"`?`" for all $i \geq 1$.

For example, f_1 is

"`What are you doing while sending "What are you doing at the end of the world? Are you busy? Will you save us?"? Are you busy? Will you send "What are you doing at the end of the world? Are you busy? Will you save us?"?`". Note that the quotes in the very beginning and in the very end are for clarity and are not a part of f_1.

It can be seen that the characters in f_i are letters, question marks, (possibly) quotation marks and spaces.

Nephren will ask the little leprechauns q times. Each time she will let them find the k-th character of f_n. The characters are indexed starting from 1. If f_n consists of less than k characters, output '`.`' (without quotes).

Can you answer her queries?

Input

The first line contains one integer q ($1 \leq q \leq 10$) — the number of Nephren's questions.

Each of the next q lines describes Nephren's question and contains two integers n and k ($0 \leq n \leq 10^5$, $1 \leq k \leq 10^{18}$).

Output

One line containing q characters. The i-th character in it should be the answer for the i-th query.

Examples

input
3 1 1 1 2 1 111111111111

output
Wh.

input
5 0 69 1 194 1 139 0 47

```
1 66
```

output

```
abdef
```

input

```
10
4 1825
3 75
3 530
4 1829
4 1651
3 187
4 584
4 255
4 774
2 474
```

output

```
Areyoubusy
```

Note

For the first two examples, refer to f_0 and f_1 given in the legend.

B. Ithea Plays With Chtholly

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

This is an interactive problem. Refer to the Interaction section below for better understanding.

Ithea and Chtholly want to play a game in order to determine who can use the kitchen tonight.

Initially, Ithea puts n clear sheets of paper in a line. They are numbered from 1 to n from left to right.

This game will go on for m rounds. In each round, Ithea will give Chtholly an integer between 1 and c, and Chtholly needs to choose one of the sheets to write down this number (if there is already a number before, she will erase the original one and replace it with the new one).

Chtholly wins if, at any time, all the sheets are filled with a number and the n numbers are in non-decreasing order looking from left to right from sheet 1 to sheet n, and if after m rounds she still doesn't win, she loses the game.

Chtholly really wants to win the game as she wants to cook something for Willem. But she doesn't know how to win the game. So Chtholly finds you, and your task is to write a program to receive numbers that Ithea gives Chtholly and help her make the decision on which sheet of paper write this number.

Input

The first line contains 3 integers n, m and c ($n, m \geq 2, 1 \leq c \leq 1000, 1 \leq n \cdot \lceil \frac{c}{2} \rceil \leq m \leq 1000, \lceil x \rceil$ means x rounded up) — the number of sheets, the number of rounds and the largest possible number Ithea can give to Chtholly respectively. The remaining parts of input are given throughout the interaction process.

Interaction

In each round, your program needs to read one line containing a single integer p_i ($1 \leq p_i \leq c$), indicating the number given to Chtholly.

Your program should then output a line containing an integer between 1 and n, indicating the number of sheet to write down this number in.

After outputting each line, don't forget to flush the output. For example:

- `fflush(stdout)` in C/C++;
- `System.out.flush()` in Java;
- `sys.stdout.flush()` in Python;
- `flush(output)` in Pascal;
- See the documentation for other languages.

If Chtholly wins at the end of a round, no more input will become available and your program should terminate normally. It can be shown that under the constraints, it's always possible for Chtholly to win the game.

Example

input
2 4 4 2 1 3
output
1 2 2

Note

In the example, Chtholly initially knew there were 2 sheets, 4 rounds and each number was between 1 and 4. She then received a 2 and decided to write it in the 1st sheet. Then she received a 1 and wrote it in the 2nd sheet. At last, she received a 3 and replaced 1 with 3 in the 2nd sheet. At this time all the sheets were filled with a number and they were non-decreasing, so she won the game.

Note that it is required that your program terminate immediately after Chtholly wins and do not read numbers from the input for the remaining rounds. If not, undefined behaviour may arise and it won't be sure whether your program will be accepted or rejected. Also

because of this, please be careful when hacking others' codes. In the sample, Chtholly won the game after the 3rd round, so it is required that your program doesn't read the number of the remaining 4th round.

The input format for hacking:

- The first line contains 3 integers n, m and c;
- The following m lines each contains an integer between 1 and c, indicating the number given to Chtholly in each round.

C. Willem, Chtholly and Seniorious

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

— Willem...

— What's the matter?

— It seems that there's something wrong with Seniorious...

— I'll have a look...

Seniorious is made by linking special talismans in particular order.

After over 500 years, the carillon is now in bad condition, so Willem decides to examine it thoroughly.

Seniorious has n pieces of talisman. Willem puts them in a line, the i-th of which is an integer a_i.

In order to maintain it, Willem needs to perform m operations.

There are four types of operations:

- $1\ l\ r\ x$: For each i such that $l \le i \le r$, assign $a_i + x$ to a_i.
- $2\ l\ r\ x$: For each i such that $l \le i \le r$, assign x to a_i.
- $3\ l\ r\ x$: Print the x-th smallest number in the index range $[l, r]$, i.e. the element at the x-th position if all the elements a_i such that $l \le i \le r$ are taken and sorted into an array of non-decreasing integers. It's guaranteed that $1 \le x \le r - l + 1$.
- $4\ l\ r\ x\ y$: Print the sum of the x-th power of a_i such that $l \le i \le r$, modulo y, i.e. $(\sum_{i=l}^{r} a_i^x) \bmod y$.

Input

The only line contains four integers n, m, $seed$, v_{max} ($1 \le n, m \le 10^5$, $0 \le seed < 10^9 + 7$, $1 \le vmax \le 10^9$).

The initial values and operations are generated using following pseudo code:

```
def rnd():

    ret = seed
    seed = (seed * 7 + 13) mod 1000000007
    return ret

for i = 1 to n:

    a[i] = (rnd() mod vmax) + 1

for i = 1 to m:

    op = (rnd() mod 4) + 1
    l = (rnd() mod n) + 1
    r = (rnd() mod n) + 1

    if (l > r):
         swap(l, r)

    if (op == 3):
        x = (rnd() mod (r - l + 1)) + 1
```

```
else:
    x = (rnd() mod vmax) + 1

if (op == 4):
    y = (rnd() mod vmax) + 1
```

Here op is the type of the operation mentioned in the legend.

Output

For each operation of types 3 or 4, output a line containing the answer.

Examples

input
10 10 7 9
output
2 1 0 3

input
10 10 9 9
output
1 1 3 3

Note

In the first example, the initial array is $\{8, 9, 7, 2, 3, 1, 5, 6, 4, 8\}$.

The operations are:

- 2 6 7 9
- 1 3 10 8
- 4 4 6 2 4
- 1 4 5 8
- 2 1 7 1
- 4 7 9 4 4
- 1 2 7 9
- 4 5 8 1 1
- 2 5 7 5
- 4 3 10 8 5

D. Nephren Runs a Cinema

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Lakhesh loves to make movies, so Nephren helps her run a cinema. We may call it No. 68 Cinema.

However, one day, the No. 68 Cinema runs out of changes (they don't have 50-*yuan* notes currently), but Nephren still wants to start their business. (Assume that *yuan* is a kind of currency in Regulu Ere.)

There are three types of customers: some of them bring exactly a 50-*yuan* note; some of them bring a 100-*yuan* note and Nephren needs to give a 50-*yuan* note back to him/her; some of them bring VIP cards so that they don't need to pay for the ticket.

Now n customers are waiting outside in queue. Nephren wants to know how many possible queues are there that they are able to run smoothly (i.e. every customer can receive his/her change), and that the number of 50-*yuan* notes they have after selling tickets to all these customers is between l and r, inclusive. Two queues are considered different if there exists a customer whose type is different in two queues. As the number can be large, please output the answer modulo p.

Input

One line containing four integers n ($1 \le n \le 10^5$), p ($1 \le p \le 2 \cdot 10^9$), l and r ($0 \le l \le r \le n$).

Output

One line indicating the answer modulo p.

Examples

input
4 97 2 3
output
13

input
4 100 0 4
output
35

Note

We use A, B and C to indicate customers with 50-*yuan* notes, customers with 100-*yuan* notes and customers with VIP cards respectively.

For the first sample, the different possible queues that there are 2 50-*yuan* notes left are AAAB, AABA, ABAA, AACC, ACAC, ACCA, CAAC, CACA and CCAA, and the different possible queues that there are 3 50-*yuan* notes left are AAAC, AACA, ACAA and CAAA. So there are 13 different queues satisfying the first sample. Similarly, there are 35 different queues satisfying the second sample.

E. Welcome home, Chtholly

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

— I... I survived.

— Welcome home, Chtholly.

— I kept my promise...

— I made it... I really made it!

After several days of fighting, Chtholly Nota Seniorious miraculously returned from the fierce battle.

As promised, Willem is now baking butter cake for her.

However, although Willem is skilled in making dessert, he rarely bakes butter cake.

This time, Willem made a big mistake — he accidentally broke the oven!

Fortunately, Chtholly decided to help him.

Willem puts n cakes on a roll, cakes are numbered from 1 to n, the i-th cake needs a_i seconds of baking.

Willem needs Chtholly to do m operations to bake the cakes.

Operation 1: $1\ l\ r\ x$

Willem asks Chtholly to check each cake in the range $[l, r]$, if the cake needs to be baked for more than x seconds, he would bake it for x seconds and put it back in its place. More precisely, for every i in range $[l, r]$, if a_i is strictly more than x, a_i becomes equal a_i - x.

Operation 2: $2\ l\ r\ x$

Willem asks Chtholly to count the number of cakes in the range $[l, r]$ that needs to be cooked for exactly x seconds. More formally you should find number of such i in range $[l, r]$, that $a_i = x$.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$).

The second line contains n integers, i-th of them is a_i ($1 \le a_i \le 10^5$).

The next m lines are the m operations described above. It is guaranteed that $1 \le l \le r \le n$ and $1 \le x \le 10^5$.

Output

For each operation of the second type, print the answer.

Examples

input
5 6 1 5 5 5 8 2 2 5 5 1 2 4 3 2 2 5 2 2 2 5 5 1 3 5 1 2 1 5 1

output
3 3 0 3

input
7 7 1 9 2 6 8 1 7 2 1 7 1 2 2 5 2 1 4 7 7 2 2 4 2 1 3 4 5 2 3 3 3 2 3 7 2

output
2 1

```
1
0
1
```

input

```
8 13
75 85 88 100 105 120 122 128
1 1 8 70
2 3 8 30
1 3 8 3
2 2 5 15
1 2 4 10
2 1 5 5
1 2 7 27
2 1 5 5
1 3 7 12
1 1 7 4
2 1 8 1
1 4 8 5
2 1 8 1
```

output

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