A. Roll the string

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Given a string consisting of only lower-case characters. You have to perform the operation on the string. The operation consists of changing all the letters of string by skipping p characters(including itself) ahead of them in alphabetical order. For eg. if s = "abcd" and p is 3 s becomes "defg". Print the final string.

Input

Given t test cases. First line of every test case consists of n the length of the string and p number of letters to be skipped. Next line consists of the string. 1 <= t <= 100 1 <= n <= 200000, 0 <= p <= 1000000

Output

For every test case print the final string on a newline.

B. Alice and Bob

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Alice gave bob two integers X and Y. Alice challenged Bob to find the minimum value of the two integers after performing certain number of operations on it. The operation goes as follows: Choose smaller of the two values, remove it from the bigger value and add it to the smaller value again. Bob is unable to calculate and wants your help to solve the problem. Help Bob find the smaller of the two values after G operations.

Input

First line consists t, the number of test cases. Next t lines consists of three space separated integers X, Y, G

1 <= t <= 200000

1 <= X, Y <= 1000000000

1 <= G <= 2000000000

Output

For every test case print the smaller of the two values on a new line after G operations.

Example

input
1 13 11 2
output
4

Note

After applying 1st operation X becomes 2 and Y becomes 22 After applying 2nd operation X becomes 4 and Y becomes 20. Since the smaller of 4 and 20 is 4 our answer is 4.

C. Max XOR Pair

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

This is an interactive problem!

Bob is playing a guessing game. Alice has secretly written up a sorted array of integers A of size n and Bob wants to guess an integer i such that $A_i \oplus A_{i+1}$ is the maximum possible. Bob can ask a maximum of 60 queries to Alice each query contains 2 integers l, r $(1 \leq l < r \leq n)$, and Alice will return the XOR of all elements of the subarray $A_l, A_{l+1}, \ldots, A_r$. It is given that $A_1 = 0$ for all test cases.

Input

There are t $(1 \leq t \leq 100)$ test cases and each test case contains a single integer n $(2 \leq n \leq 10^6)$ the size of the hidden array as input.

Interaction

For at most 60 queries output 3 space-separated integers

$1\ l\ r$

denoting query for XOR of the subarray of the array A from index l to r inclusive. The judge will output a single integer X the answer of the query if it is a valid query and the number of queries asked is less than 60 otherwise it will return -1. If -1 is received you have received the wrong answer verdict and should terminate immediately for expected results.

Once you have the answer you can print it on a new line as

$2\ i$

the integer i denotes the answer, the index such that $A_i \oplus A_{i+1}$ is maximum. You can print this query only once per test case and this doesn't count in the 60 queries count. Interactor will not output anything for this query.

If there are multiple correct answers print any.

After printing a query do not forget to output the end of line and flush the output. Otherwise, you will get Idleness limit exceeded. To do this, use:

- fflush(stdout) or cout.flush() in C++;
- System.out.flush() in Java;
- flush(output) in Pascal;
- stdout.flush() in Python;
- see the documentation for other languages.

Example

input
2 7 0 2 6 7 8 9 10 5 0 2 4 5 6
output
4 2

D1. Can You Beat the Dragons ?

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are a very famous dragon warrior, Your current strength is S.

You have decided to go on a war with the dragons. To kill the i^{th} dragon, You need to have at least s_i units of strength; after you kill this dragon, your strength will be added by ego_i (your strength will get added by ego_i) (ego_i can be positive or negative). Your strength should not fall below zero because then you will be dead.

Is it possible to kill all of the dragons? Formally, you have to check if such a sequence of the war exists, that you have enough strength before starting each fight, and you have non-negative strength after completing each fight.

In other words, you have to check that there exists such a sequence of the fights in which you will complete them, so you have enough strength before starting each fight, and have non-negative strength after completing each fight.

Input

The first line of the input contains two integers n and S (1≤n≤200,1≤S≤100000), the number of dragons and your initial strength.

The next n lines contain details about each fight with the dragon. The i^{th} project is represented as a pair of integers s_i and ego_i ($1 \le s_i \le 100000$, $-1000 \le b_i \le 1000$) the strength required to fight with the i^{th} dragon and the strength change after the fight.

Output

Print "Ha" or "Na" i.e. if you can beat all dragons print "Ha" else print "Na".

Example

input
3 4 4 6 10 -2 8 -1
output
Ha

D2. Can You still beat the Dragons ?

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are a very famous dragon warrior, Your current strength is S.

You have decided to go on a war with the dragons. To kill the i^{th} dragon, You need to have at least s_i units of strength; after you kill this dragon, your strength will be added by ego_i (your strength will get added by ego_i) (ego_i can be positive or negative). Your strength should not fall below zero because then you will be dead.

You can choose the order in which you fight with each dragon. Moreover, you can even skip some fights with dragons.

You want to be the bravest warrior So you want to choose the subset of fights having a maximum possible size and the order in which you will complete them, so you have enough strength before starting each fight, and have non-negative strength after completing each fight.

Your task is to calculate the maximum possible size of such a subset of fights.

Input

The first line of the input contains two integers n and S (1≤n≤200,1≤S≤100000), the number of dragons and your initial strength.

The next n lines contain details about each fight with the dragon. The i^{th} project is represented as a pair of integers s_i and ego_i ($1 \le s_i \le 100000$, $-1000 \le b_i \le 1000$) the strength required to fight with the i^{th} dragon and the strength change after the fight.

Output

Print one integer — the size of the maximum possible subset can be an empty one, that you can choose.

Example

input

```
5 20
45 -6
34 -15
10 34
1 27
40 -45
```

output

```
5
```

F. Switch Tree

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

There is a circuit of bit storage devices in the form of a tree, i.e. each node has a bit value 0 or 1. You can perform 2 operations on it.

1. You set the bit of the selected node and all nodes in its subtree.
2. You clear the bit of the selected node and all its ancestors.

Initially, all bits are cleared (value=0).

Given q queries where each query contains 2 space-separated integers d, k where d denotes the query type and k is the index of the node. The types of queries are:

1. $d = 1$, Operation 1 is performed on node k, No output from user required.
2. $d = 2$, Operation 2 is performed on node k, No output from user required.
3. $d = 3$, Print value of bit of node k

Note: The queries are not independent, the changes made by a query are reflected in all the following queries.

Input

The first line of input contains a single integer n $(1 \leq n \leq 10^5)$ the number of nodes in the tree.

Next $n - 1$ lines contain 2 space-separated integers u, v $(1 \leq u, v \leq n)$ denoting an edge between u and v.

The next line contains a single integer q $(1 \leq q \leq 10^5)$ denoting the number of queries.

Next q lines contain 2 space-separated integers d, k $(1 \leq d \leq 3),\ (1 \leq k \leq n)$ denoting the queries.

Output

For each query of type 3 print a single integer (0 or 1) in a new line denoting the answer to the query.

Example

input

```
5
1 2
5 1
2 3
4 2
12
1 1
2 3
3 1
3 2
3 3
3 4
1 2
2 4
3 1
3 3
3 4
3 5
```

output

```
0
0
0
```

```
1
0
1
0
1
```

G. Seong Gi-Hun

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

Frontman gave Seong Gi-Hun a task. He has to find the number of arrays that satisfy the following conditions:

- $0 \leq A[i] \leq M$
- $A[0] + A[1] + A[2] + ... + A[N] = M$
- $A[0]$ | $A[1]$ | $A[2]$ | ... | $A[N] = K$

As Seong Gi-Hun is weak at studies he wants you to calculate the answer.

Input

The single line contains three space-separated integers N, M, K $(1 \leq N \leq 1000)$ $(1 \leq M, K \leq 5000)$

Output

Print the required answer modulo $10^9 + 7$

Example

input
2 3 3
output
4

H. Cube Game

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

There are N cubes in this game. All cubes are connected with each other using $N - 1$ tunnels. Each cube has beauty assigned to it. For cube i this beauty is $A[i]$.

Let's assume you are at cube x, then you can jump to cube $y(x! = y)$ for which $A[y] - d(x, y)$ is maximum. here $d(x, y)$ is the number of edges between x and y.

You are given Q queries in the form of starting cube and number of jumps, You have to answer the cube you are standing at after performing exactly K jumps.

Note: If there are multiple cubes that have maximum value then jump to the cube which has the smallest index.

Input

First Line contains a single integer N the number of nodes.

Second-line contains N integer denotes the beauty of i^{th} cube.

next $N-1$ line contains two integers u and v, with the meaning that there is a tunnel between cubes u and v.

The next line will contain a single integer - Q - the number of queries

Next, Q lines will contain two integers $start$(starting cube) and K(number of jumps)

$1 \leq N, Q \leq 10^5$

$1 \leq u, v, start \leq N$

$1 \leq K \leq 10^9$

Output

For each query answer the index of the cube you are standing at after jumping exactly K times.

Example

input
5 1 1 3 2 4 1 2 1 3 2 4 2 5 1 1 4
output
5