Safe Steps

Percy is going to Olympus to return the stolen lightening bolt. To his surprise, each of the steps that lead to Zeus's palace are marked with N bits incrementally starting from 0 to 2^N, 1 <= N <= 16. Now, to reach the top safely Percy can only use the steps that have exactly H bits set (he has to step over any other steps), 1 <= H <= N <= 16. Help Percy find the sequence of all the steps on which he can step or else Poseidon and Zeus will destroy the world of mortals if the lightening bolt is not returned.

Input

The input consists of N and H separated by white space on a single line.

Output

Print the numbers of steps one per line from (numerically) smallest to largest. The output should end with a newline.

Example 1

```
Input:

4 1

Output:

0001
0010
0100
1000
```

Example 2

```
Input:

4 2

Output:

0011
0101
0110
1001
1010
1100
```

Desolate Number

Given two integers A and B, a desolate number N is defined as follows:

- N is an (A+B)-bit integer;
- the binary representation of N has exactly A 1's and B 0's (leading zeroes are ok);
- N has the maximum number of 1's adjacent to at least one 0 in its binary representation.

Your task is to find the smallest desolate number N given the values of A and B.

Input

The input consists of a single line, containing two non-negative integers A and B ($1 <= A + B <= 50$).

Output

Print one line, containing the smallest desolate number.

Example 1

```
Input:
1 1

Output:
1
```

Since we have two bits and one of them has to be zero and the other one has to be one, then the two options are `01` and `10`. They both have the same number of 1-bits adjacent to the 0-bits. But the first one is smaller, so the answer is 1.

Example 2

```
Input:
4 3

Output:
45
```

45 is binary using 7 bits is `0101101`. In this sequence every 1-bit is adjacent to at least one 0 bit so this constraint is maximized. The only way to make this value smaller is to shift some of the 1-bits to the right, but this would decrease the number of 1-bits adjacent to the 0-bits.

Example 3

```
Input:
4 1

Output:
23
```

23 in binary using 5 bits is `10111`. With a single zero available at most two 1-bits can be adjacent to a 0-bit. If we move the 0-bit any further to the right, the value would increase.

HINT: You do not need to try every possible combination of the bits to solve this problem.

Laughing Monsters

The world has been attacked by Laughing Monsters. They camouflage as humans at movie theaters. Anybody sitting within D feet from a Laughing Monster will get infected and will have to endure uncontrollable laughter attacks for the next week. Moreover, once infected, these people will infect other humans sitting within D feet of them.

The movie theater owners need to be able to provide information about anybody who might have been exposed to this infection if the authorities identify a particular person to be really a camouflaged Laughing Monster.

Your job is to write a program that will identify the groups of theater guests who are in risk of infection if one of them is identified as a Laughing Monster.

Two theater guests belong to the same group if distance between their seats is less than or equal to D feet.

Input

The first line contains the number of theater guests `N` (`0 <= N <= 1,000`) and the distance `D` (a real number `0.0 <= D <= 1,000.00`). Next `N` lines have a pair of real coordinates `X Y` (where `-1,000.00 <= X, Y <= 1,000.00`) for location of each guest in the theater.

All real numbers in the input will have at most 2 (two) digits after a decimal point.

Output

Output the number of distinct groups.

Example 1

```
Input:
5 1.5
1.0 0.1
2.0 0.0
5.0 0.2
6.0 0.4
3.0 -0.1

Output:
2
```

Example 2

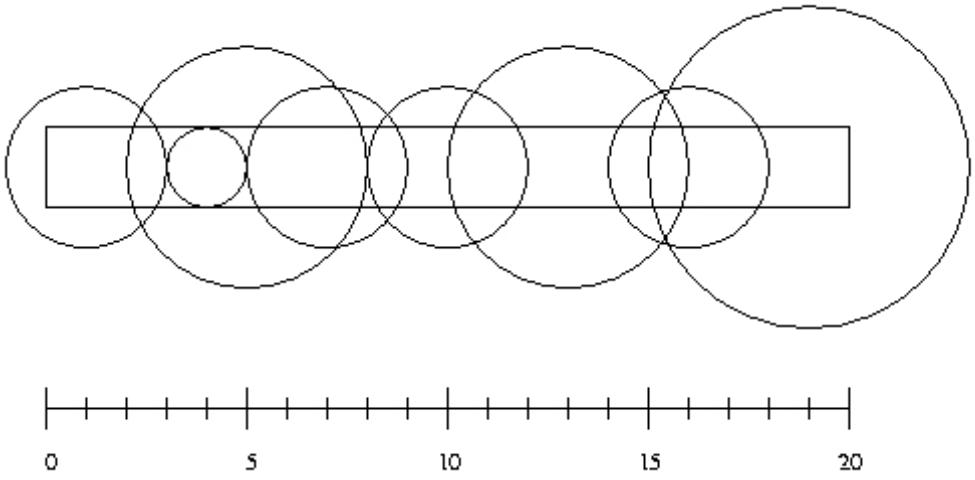
```
Input:
3 4.0
121.12 254.06
645.04 301.85
912.49 568.96

Output:
3
```

Sprinklers

Farmer John has a large field, and he is thinking of planting sweet corn in some part of it. After surveying his field, FJ found that it forms a horizontal strip l meters long and w meters wide.

There are n sprinklers installed at the horizontal center line of the strip. For each sprinkler you are given its radius of operation and its position as the distance from the left end of the center line. Farmer John wants to plant sweet corn in the whole field, but he wishes to turn on as few sprinklers as possible to save the water.

Find the minimum number of sprinklers that need to be turned on in order to water the entire field.



Input

The first line contains integer numbers n, l and w with 1 <= n <= 10,000, 1 <= l <= 10,000,000, and 1 <= w <= 100. The next n lines contain two integers giving the position x (0 <= x <= l) and radius of operation r (1 <= r <= 1000) of a sprinkler.

The picture above illustrates the first case from the sample input.

Output

Output one integer followed by a newline: the minimum number of sprinklers needed to water the entire strip.

If it is impossible to water the entire strip, output -1.

Example 1 (shown in the image above)

```
Input:
8 20 2
5 3
4 1
1 2
7 2
10 2
13 3
16 2
19 4

Output:
6
```

Example 2

```
Input:
3 10 1
```

```
3 5
9 3
6 1

Output:
2
```

Example 3

```
Input:
3 10 1
5 3
1 1
9 1

Output:
-1
```

Batch Processing

Since Light Kingdom did not pay you salary for a whole year, you decided to leave and work for Conflict Empire. Now, you are again an operator of a super computing center and in control of M nodes.

One day, a research institute from Conflict Empire submitted N computational tasks numbered from 1 to N. Given the running time needed for each task, you are to distribute the tasks among the available nodes such that the node with heaviest workload completes as early as possible. Restriction: every node must process at least one task and must process a contiguous subsection of tasks. That is, you need to find a sequence 0 = L0 < L1 < ... < LM-1 < LM = N where the i-th node processes tasks Li-1+1, Li-1+2, ..., Li.

Input

The first line of the input contains two integers N and M (1 <= M <= N <= 500), representing the number of tasks and the number of nodes you control, respectively. The second line contains N integers Ti (1 <= Ti < 10,000,000), representing the time needed to complete each task.

Output

Print one line containing the input T1, T2, ..., TN divided into M parts such that the maximum sum of a single part is minimized. You should use character '/' to separate the parts and there must be a space character between every numbers or '/'s.

If there is more than one solution, print the one that minimizes the sum of the first part, then the second part and so on.

Example 1

```
Input:
9 3
100 200 300 400 500 600 700 800 900

Output:
100 200 300 400 500 / 600 700 / 800 900
```

Example 2

```
Input:
5 4
100 100 100 100 100

Output:
100 / 100 / 100 / 100 100
```