Question 0 Enumeration (50 Marks)

Program Name:	PROGRAM0.EXE
Input:	Standard Input
Output:	Standard Output
Maximum Execution Time:	1 second

Write a program to read an integer N and output N^2 integers in specified format.

Input

The input contains an integer N ($1 \leq N \leq 10$).

Output

The output contains a triangle with its base at the left, of length 2N-1. Every integer will be greater than the integer above by 1 and it will be greater than the integer on the left.

Sample Input

```
4
```

Sample Output

```
1
2 8
3 9 13
4 10 14 16
5 11 15
6 12
7
```

Question 1 Squares Dartboard (100 Marks)

Program Name:	PROGRAM1.EXE
Input:	Standard Input
Output:	Standard Output
Maximum Execution Time:	1 second

Organization of Intelligence (OI) introduces a new game to its members. The following are the game rules:

1. The player shoots at the dartboard.
2. The dartboard is divided into several areas, where the border lines are:
   Type A: a square with its sides parallel to the x-axis and the y-axis
   Type B: a Type A square which is rotated 45 degrees
   All squares have their centre at the origin and the border lines do not intersect or touch each other.
3. Each square contains a score. The score of a shot is the score of the square with least area that contains the dart.
4. The player gets 0 score if the shot is right on a border line or it is outside all border lines.
5. The total score is the sum of scores obtained in all shots.

Given the dartboard and the position of the shots, write a program to output the total score of the player.

Input

The first line contains two integers *N* and *M*, which indicates the number of border lines and the number of shoots made respectively.
Each of the following *N* lines contains two integers d_i , p_i and one character t_i , separated by a single space, which are the radius, score and type of the border line. If the square is type A, its radius is the distance of the sides to the origin and t_i is an uppercase "A". If the square is type B, its radius is the distance of the corners to the origin and t_i is an uppercase "B".
Each of the following *M* lines contains two integers x_j and y_j, separated by a single space, which is the x and y coordinate of the jth shot.

Output

The output consists of one integer, the total score of the player.

Sample Input

```
2 3
9 10 A
4 23 B
0 1
-1 3
6 -7
```

Sample Output

```
33
```

Constraints

In all the test cases,

- $1 \le N \le 3{,}000$
- $1 \le M \le 100{,}000$
- $1 \le p_i, d_i \le 10{,}000$
- $-30{,}000 \le x_j, y_j \le 30{,}000$

In 50% of the test cases,

- $1 \le N, M \le 1{,}000$

Question 2 Igloo (100 Marks)

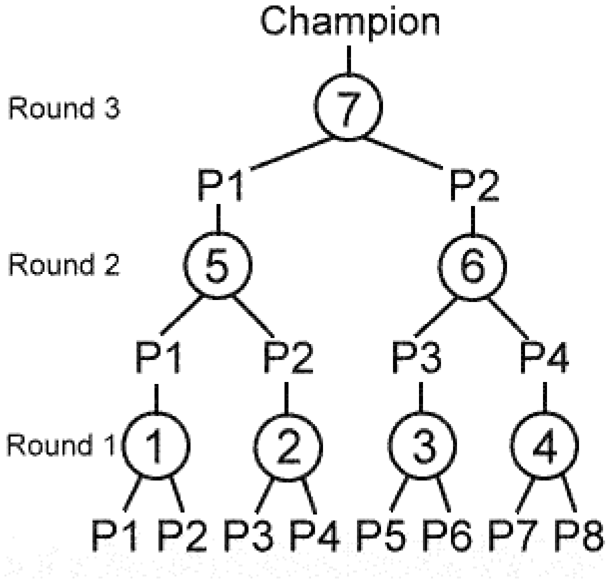
Program Name:	PROGRAM2.EXE
Input:	Standard Input
Output:	Standard Output
Maximum Execution Time:	1 seconds

The HiKing On IglOo Contest (HKOIOC) is an annual competition in Antarctica organized by penguins. Penguins in different igloos compete to gain glory for their tribes.
There are 2^N penguins in the competition with contestant number $1...2^N$. Initially the penguins are labeled according to their contestant numbers. The label of a penguin may change during the competition, but the contestant number will not change.

The competition consists of N rounds of matches. In a match, two penguins races on the top of an igloo. The penguin with larger height wins the match. In case of a tie, the penguin with smaller contestant number wins. The loser is eliminated from the competition.

The competition is a single-elimination tournament. For simplicity, the penguin labeled i is denoted by Pi. In the first round, P1 races against P2, and the winner is relabeled 1. Then P3 races against P4, and the winner is relabeled 2. Then P5 races against P6, and the winner is relabeled 3, and so on. After the first round, there are 2^{N-1} penguins left, which are labeled $1...2^{N-1}$. The following rounds proceed just like the first round. After the N^{th} round, there is only one penguin left, which is the champion of HKOIOC.

The following figure shows the case when there are 8 penguins. Each circle represents a match. Numbers in the circles indicate the order in which the matches are conducted.



Each igloo sends a number of penguins to the competition. Penguins in the same igloo share the same height. The penguins of the same igloo register for the competition together. Therefore their contestant numbers are consecutive.

In order to add excitement to the competition, the organizing committee may invite the penguins from the Igloo of Tux to join the competition. The penguins from the Igloo of Tux are characterized by heights 600cm, which overshadows many other penguins. Of course the organizing committee would not allow such monsters to win the competition. Therefore they add a rule stating that if only one penguin which is not from the Igloo of Tux remains at any moment during the competition, then the penguin automatically becomes the champion.

Given the height of penguins and the number of penguins sent of each igloo, write a program to output the contestant number of the champion.

Input

The first line contains an integer *M*, the number of igloos.

The ith line in the next *M* lines contains two integers h_i and a_i, where h_i is the height (in cm) of penguins of the ith igloo, and a_i is the number of penguins sent from the igloo. The igloos are sorted in increasing order of the contestant numbers of its penguins. A penguin has height equals 600cm only if it comes from the Igloo of Tux.

You may assume that the sum of a_i (i.e. the number of penguins) is a power of 2, the number 600 appears at most once in h_i, and there is at least one igloo with height of penguins not equal to 600.

Output

The output consists of one line with an integer, the contestant number of the champion.

Sample Input

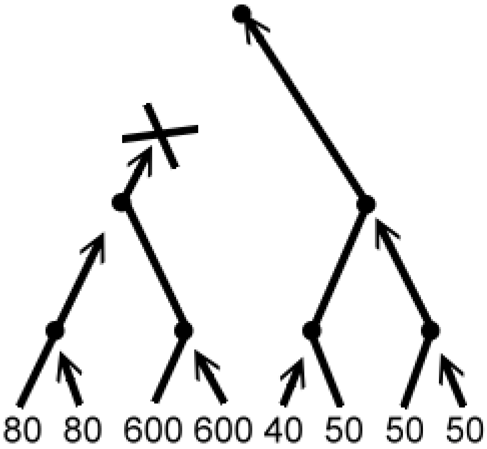
```
4
80 2
600 2
40 1
50 3
```

Sample Output

```
6
```

Explanation

The figure below describes how the penguins advance in the competition. After round 2, only one penguin with weight not equal to 600cm is left, and thus becomes the champion according to the rule.



Constraints

In all the test cases,

- $1 \leq M \leq 100{,}000$
- $1 \leq a_{\mathrm{i}} \leq 5{,}000$
- $1 \leq h_{\mathrm{i}} \leq 1{,}000$

In 50% of the test cases,

- $1 \leq M \leq 100$
- $1 \leq a_{\mathrm{i}} \leq 200$

In 20% of the test cases,

- $1 \leq h_{\mathrm{i}} \leq 500$

Question 3 Flexibility (100 Marks)

```
Program Name:                   PROGRAM3.EXE
Input:                          Standard Input
Output:                         Standard Output
Maximum Execution Time:         1 second
```

The HiKing On IglOo Contest (HKOIOC) is finished successfully this year. There are always some losers trying their best to upgrade themselves and aiming at winning the game in the next year. One of them is Tux.

One facet affecting their flexibility is their weight. Penguins with different weights should have different strategies during the contest. Therefore, it is important for penguins to know their weights.

However, there are only some masses and traditional balances in the penguin society. Tux is given a traditional balance and N masses with weights A_1, A_2,..., A_N, in ascending order. Note that those masses are well designed so that $A_{i+1} \geq 3A_i$.

Tux, with weight W, will stand on the left side of the balance. You can put some of those N masses on the left hand side or right hand side in order to balance the balance. Each mass can be used at most once. Write a program to help Tux to balance the balance.

Input

The first line of the input contains 2 integers, W and N. The second line contains N integers, A_1, A_2,..., A_N, which are the weights of the masses.

Output

The first line contains the number of masses put on the left hand side and then the weight of each mass on the left hand side in ascending order.

The second line contains the number of masses put on the right hand side and then the weight of each mass on the right hand side in ascending order.

If there is more than one solution, you may output any one of them.

If it is impossible to balance the balance, output one line "Impossible" (without quotes).

Sample Input 1

```
16 5
1 3 10 30 99
```

Sample Output 1

```
3 1 3 10
1 30
```

Sample Input 2

```
15 5
1 3 10 30 99
```

Sample Output 2

```
Impossible
```

Constraints

In all the test cases,

- $1 \leq N \leq 35$
- $1 \leq A_{\mathrm{i}} \leq 2^{60}$

In 50% of the test cases,

- $1 \leq N \leq 10$
- $1 \leq A_{\mathrm{i}} \leq 2^{30}$

Extra Note

To handle integers larger than $2^{32} - 1$, contestants using C/C++ need to use the data type "`long long`" (without quotes). The following is an example for "`long long`" input and output.

```
#include "stdio.h"
int main(){
      long long x;
      scanf("%I64d", &x);
      printf("%I64d\n",x);
      return 0;
}
```

Contestants using Pascal need to use the data type "`int64`" (without quotes). Please note that "`int64`" is not an ordinal data type. The following is an example for "`int64`" input and output.

```
var x : int64;
begin
      readln(x);
      writeln(x);
end.
```

Question 4 OI Football Team (100 Marks)

```
Program Name:                  PROGRAM4.EXE
Input:                         Standard Input
Output:                        Standard Output
Maximum Execution Time:        1 seconds
```

In OI Football Team, there are N players whose numbers are 1, 2, 3…N respectively. Since both long pass and short pass are very important techniques when playing football, the coach of OI Football Team creates an interesting method to train the players. Each player stands on a point of the football field and will not move during the training. The distance between two points with coordinates (x_1, y_1) and (x_2, y_2) is given by:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The process starts at player A. Player A kicks the ball to player B ($1 \le B \le N$, $B \neq A$) with the **greatest** distance to player A. Then player B kicks the ball to player C ($1 \le C \le N$, $C \neq B$), with the **least** distance to player B. Player C then passes the ball to the player with the **greatest** distance to him, and so on. Note that if there is more than one possible player to pass to, the player keeping the ball will pass the ball to the player whose number is smaller.

The players may feel tired after training for a long time. However, if all players go to rest at the same time, the coach thinks that it wastes time. Then he thinks of a method. That is, after the ball is passed for K times, the player keeping the ball passes the ball to the teammate with the least distance to him, then leaves and takes a rest. The process then starts from beginning until there is only one player left.

Assume at the beginning of the training, player 1 holds the ball. Write a program to help the coach find the last player.

Input

The first line contains 2 integers, N and K. The ith line of the following N lines contains two integers, X_i and Y_i, representing the coordinates of player i.

Output

The output contains one integer only, the number of the last player.

Sample Input

```
5 3
0 0
2 4
1 7
3 2
6 5
```

Sample Output

```
2
```

Explanation

The following sequence shows the players who keep the ball as the training proceeds:
1 → 5 → 2 → 1 (leave) → 2 → 5 → 2 → 5 (leave) → 2 → 3 → 2 → 3 (leave) → 2 → 4 → 2 → 4 (leave) → 2

Constraints

In all the test cases,

- $2 \leq N \leq 200$
- $1 \leq K \leq 50{,}000$
- $0 \leq X_i, Y_i \leq 10{,}000$

In 50% of the test cases,

- $2 \leq N \leq 50$
- $1 \leq K \leq 1{,}000$
- $0 \leq X_i, Y_i \leq 1{,}000$