

Problem A

Blackjack

Time limit: 2 seconds

As the saying goes, “if you are hesitant, you will lose the game; if you are decisive, you will die in vain.”

Calabash doesn't want to lose the game, neither to die in vain. But, this is not important, because he doesn't have enough money to buy the game.

Fortunately, Roundgod, the banker of a casino, gives him a chance to fulfill his dream. Roundgod promises to buy him the game if he could win in one round of Blackjack. The Blackjack involves a deck of playing cards, each has some points on it. It is carried out as follows: Roundgod gives a number a and tells Calabash the number of points on each of the cards. Then, Calabash begins to draw cards, according to the following procedure:

1. the deck of all cards is shuffled uniformly at random;
2. Calabash could repeatedly draw a card from the deck (he immediately knows the number of the points on the card after drawing the card), or declare to stop drawing at any moment;
3. if, after drawing any card, the total number of points on cards Calabash drawn exceeds a limit b , he loses immediately;
4. otherwise, after Calabash stops drawing, he wins if and only if the total number of points on cards he drew is greater than a .

Calabash wants to know the probability to win if he plays optimally so that he can obtain his favorite game. Please help him to calculate the probability.

Input

There are two lines in the input. The first line consists of three integers n, a, b ($1 \leq n \leq 500, 1 \leq a < b \leq 500$), denoting the number of cards in the deck, the number given by Roundgod, and the limit to the total number of points, respectively.

The second line contains n integers x_1, x_2, \dots, x_n ($1 \leq x_i \leq 500$), denoting the numbers of points of these cards. It is guaranteed that $\sum_{i=1}^n x_i > b$.

Output

Output the answer within an absolute error of no more than 10^{-6} .

Sample Input 1

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

Sample Output 1

```
0.100000000000
```

Sample Input 2

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

Sample Output 2

```
0.45000000000000
```

Problem B

Coffee Chicken

Time limit: 1 second

Dr. JYY has just created the Coffee Chicken strings, denoted as $S(n)$. They are quite similar to the Fibonacci soup — today's soup is made by mingling yesterday's soup and the day before yesterday's soup:

- $S(1) = \text{"COFFEE"};$
- $S(2) = \text{"CHICKEN"};$
- $S(n) = S(n-2) :: S(n-1)$, where $::$ denotes string concatenation.

The length of $S(n)$ quickly gets out of control as n increases, and there would be no enough memory in JYY's game console to store the entire string. He wants you to find 10 contiguous characters in $S(n)$, starting from the k th character.

Input

The first line of input is a single integer T ($1 \leq T \leq 1000$), denoting the number of test cases. Each of the following T lines is a test case, which contains two integers n, k ($1 \leq n \leq 500, 1 \leq k \leq \min\{|S(n)|, 10^{12}\}$). $|S|$ denotes the length of string S .

Output

For each test case, print the answer in one line. If there are no enough characters from the k th character, output all characters until the end of the string.

Sample Input 1

```
2
3 4
2 7
```

Sample Output 1

```
FEECHICKEN
N
```

This page is intentionally left blank.

Problem C

Gifted Composer

Time limit: 8 seconds

Acesrc is a gifted composer. He likes writing tuneful and melodic songs. Every song he writes can be viewed as a sequence of musical notes, and each musical note represents the pitch and the duration of the sound. In this problem, we consider only the following seven primary pitches

do re mi fa sol la si

and the duration of each note is one unit time. Hence, there are only seven types of notes, and we may use the pitch name to represent a note.

Acesrc composes a song in the following way. Initially, the sequence of notes is empty. Every day, he inserts a new note at the beginning or at the end of the sequence, until the song is done.

Acesrc particularly likes songs with repetitions. For a song with n musical notes, we say the song has a repetition of length k ($1 \leq k \leq n$), if the song can be partitioned into one or more identical sections with k notes, optionally followed by an incomplete section, which is an initial part of a complete section. For example, do re do re do can be partitioned into do re | do re | do, so it has a repetition of length 2; similarly, do re mi do re mi has a repetition of length 3, and do re do re mi has a repetition of length 5.

Acesrc wants to know, after he adds a note each day, the number of different lengths of repetitions the song has. Can you help him?

Input

The first line of input consists of a single line n ($1 \leq n \leq 10^6$), the number of days Acesrc uses to compose the song. The i th of the remaining n lines contains a character a ($a \in \{p, a\}$) (where p denotes prepend, i.e., inserting at the beginning, and a denotes append, i.e., inserting at the end) and a string s ($s \in \{\text{do, re, mi, fa, sol, la, si}\}$), representing the action Acesrc takes in the i th day.

Output

Output n lines. The i th line should be a single integer, denoting the answer for the i th day.

Sample Input 1

```
5
a do
p re
a re
a do
p do
```

Sample Output 1

```
1
1
2
2
3
```

Sample Input 2

```
5
a re
a do
a re
p do
a mi
```

Sample Output 2

```
1
1
2
2
1
```

Problem D

Han Xin and His Troops

Time limit: 1 second

His majesty chatted with Han Xin about the capabilities of the generals. Each had their shortcomings. His majesty asked, "How many troops could I lead?" Han Xin replied, "Your highness should not lead more than 100000." His majesty said, "And what about you?" "For your humble servant, the more the merrier!" said Han Xin.

—Records of the Grand Historian

Han Xin was a military general who served Liu Bang during the Chu-Han contention and contributed greatly to the founding of the Han dynasty. Your friend, in a strange coincidence, also named Han Xin, is a military general as well.

One day you asked him how many troops he led. He was reluctant to tell you the exact number, but he told you some clues in certain form, for example:

- if we count the troops by threes, we have two left over;
- if we count by fives, we have three left over;
- if we count by sevens, two are left over;
- ...

You wonder the number of his troops, or he was simply lying. More specifically, you would like to know the minimum possible number of troops he leads, and if the minimum number is too large, then you suspect he was lying.

Input

The first line of input contains two integers n, m ($1 \leq n \leq 100, 1 \leq m \leq 10^{18}$), the number of clues he told you and the threshold that you think he was probably lying.

The remaining part of the input are n lines, specifying the clues, Each line consists of two integers a, b ($0 \leq b < a < 10^5$), representing a clue that b troops are left over if we count them by a 's.

Output

If there does not exist a non-negative number of troops satisfying all clues, print he was definitely lying; otherwise, if the minimum non-negative possible number of troops is greater than m , print he was probably lying; otherwise, print the minimum non-negative number.

Sample Input 1

```
3 100
3 2
5 3
7 2
```

Sample Output 1

```
23
```

Sample Input 2

```
4 10000
12 7
15 2
30 5
8 6
```

Sample Output 2

```
he was definitely lying
```

Sample Input 3

```
2 10
11 3
19 7
```

Sample Output 3

```
he was probably lying
```


Problem E

Hilbert Sort

Time limit: 6 seconds

The Hilbert curve, invented by the German mathematician David Hilbert, is one of the most famous fractal curves. The i -th Hilbert curve gives a sequential ordering of the cells in a $2^i \times 2^i$ grid.

The formal definition of Hilbert curves is recursive. The i -th Hilbert curve can be formed by connecting four $(i - 1)$ -th Hilbert curves, in the following way:

1. partition the $2^i \times 2^i$ grid into four $2^{i-1} \times 2^{i-1}$ quadrants;
2. reflect an $(i - 1)$ -th Hilbert curve across the main diagonal (from top left to bottom right), and place it in the upper left quadrant;
3. place two copies of the $(i - 1)$ -th Hilbert curve in the lower left and lower right quadrants, respectively;
4. reflect an $(i - 1)$ -th Hilbert curve across the secondary diagonal (from top right to bottom left), and place it in the upper right quadrant.

The first three Hilbert curves are shown below.

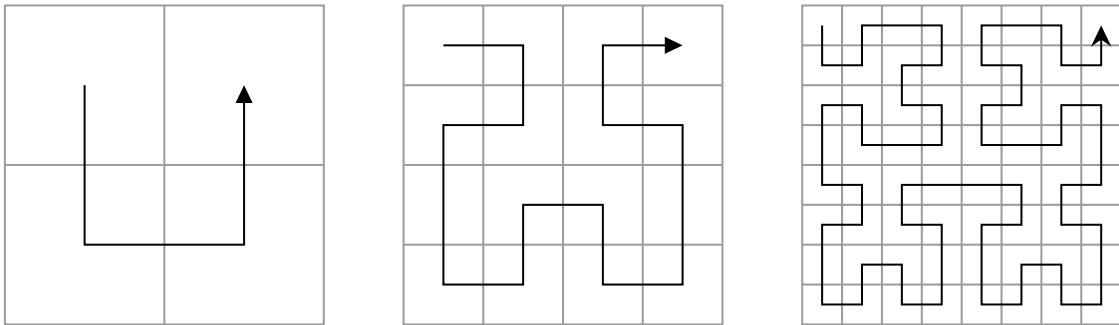


Figure 1: The 1st, 2nd and 3rd Hilbert curves.

The Hilbert sort, just as the name implies, is to sort a set of the cells according to the Hilbert curve. A cell is ranked before another if it is encountered earlier when walking along the Hilbert curve. The Hilbert sort is widely used in many areas, because such sort maps a set of 2-dimensional data points into 1-dimensional space, with spatial locality preserved fairly well.

Give you an integer k and a set of cells in the $2^k \times 2^k$ grid, please sort them in the order induced from the k -th Hilbert curve.

Input

The first line contains two integers n, k ($1 \leq n \leq 10^6, 1 \leq k \leq 30$), the number of cells to sort and the order of the Hilbert curve.

The next n lines, each containing two integers x_i, y_i ($1 \leq x_i, y_i \leq 2^k$), denoting the cell in the x_i -th row and the y_i -th column. All cells in the input are distinct.

Output

Output the coordinates of the sorted cells in n lines.

Sample Input 1

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

Sample Output 1**Sample Input 2**

4 1	1 1
1 1	2 1
1 2	2 2
2 1	1 2
2 2	

Sample Output 2

Problem F

Popping Balloons

Time limit: 5 seconds

Recently, an interesting balloon-popping game can be commonly found near the streets. The rule of this game is quite simple: some balloons are tied to cells of a lattice, and you are allowed to throw some darts to prick the balloons. The more balloons you pierce, the more incredible prize you will get.

Probably because too many people have got bored with this childish game, a new variation of the game has appeared. In this variation, you should prick the balloons by shooting an air rifle instead of throwing darts. You can make six shots in total, three horizontally and three vertically. If you shoot horizontally (vertically, resp.), all balloons in the row (column, resp.) are popped. In order to increase the difficulty, there is one restriction imposed on your shots: the distances between any two adjacent horizontal shots and any two adjacent vertical shots must all equal to a given number r . The problem is, what is the maximum number of balloons you can pop by making three horizontal and three vertical shots, under the aforementioned restriction.

Input

The first line of input is two integers n, r ($1 \leq n, r \leq 10^5$), the number of balloons and the distance between any two adjacent horizontal or vertical shots.

The remaining part of the input is n lines, specifying the positions of the balloons. Each of these lines contains two integers x, y ($0 \leq x, y \leq 10^5$), denoting a balloon positioned at (x, y) . Note that multiple balloons may lie in the same position.

Output

Output the answer as a single integer in one line.

Sample Input 1

```
7 1
0 0
1 1
2 2
3 3
4 4
5 5
7 8
```

Sample Output 1

```
6
```

Sample Input 2

```
5 2
0 10
2 3
5 7
9 6
2 3
```

Sample Output 2

```
4
```

Problem G

Road Construction

Time limit: 3 seconds

The mayor of Byteland is planning to build a new road across the town. Constructing new roads may facilitate transportation and stimulate economic development; the negative aspects include, the noise that may affect the residents nearby, and the nonnegligible impacts on the local ecological environment.

You have been required to draft a road construction plan. In your plan, you can model the residents of the town as points, and the road as a straight line of infinite length. Your plan should satisfy the following conditions:

1. The road should not pass through any resident of the city;
2. The numbers of residents on both sides of the road are equal;
3. The minimum of the distances of the residents to the road is maximized.

The following figure depicts the optimal road construction plan for the first sample test data.

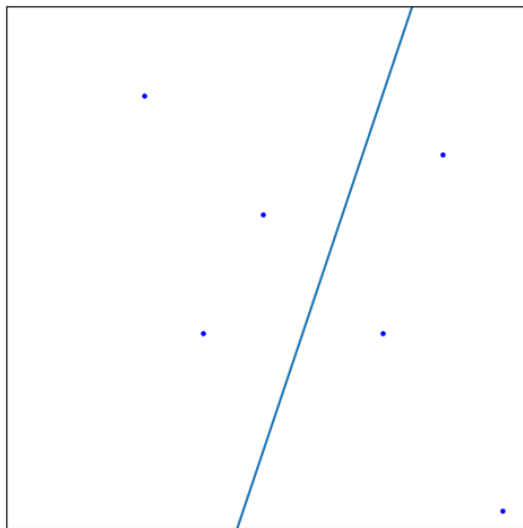


Figure 2: The optimal road construction plan for the first sample data.

Since there are too many residents in Byteland, you believe it is impossible to find such a construction plan manually. It's your time to write a program and find the optimal plan.

Input

The input starts with a line containing an even integer n ($2 \leq n \leq 300$) indicating the number of residents in Byteland. It is then followed by n lines, each containing two integers x, y ($|x|, |y| \leq 10^6$), the Cartesian coordinates of a resident. No two residents share the same coordinates.

Output

Print the minimum of the distances of the residents to the road you plan to build. Your answer should have an absolute or relative error of no more than 10^{-6} .

Sample Input 1

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

Sample Output 1

```
1.264911064067
```

Sample Input 2

```
2
1 3
2 5
```

Sample Output 2

```
1.118033988750
```

Problem H

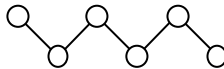
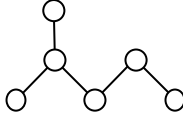
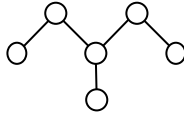
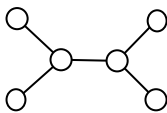
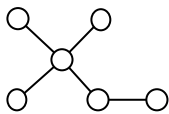
Stammering Chemists

Time limit: 3 seconds

Like many other organic chemists, you are busy in synthesizing, separating, purifying and identifying new chemical compounds all year round. Getting bored with traditional substance isolation and identification methods, such as column chromatography, a bold idea comes to your mind: *Computer Assisted Molecule Identification* (CAMI). With this technology, millions of chemists can be freed from arduous manual labor. How promising!

Currently, you are developing a simplified version of CAMI that works for hydrocarbons (organic compounds consisting of hydrogen and carbon only). All hardware-side technologies are done, and you are just going to write the program. The sensor reports all carbon atoms as well as the chemical bonds between them. This naturally forms a graph, where carbon atoms are vertices, and the bonds are edges. Each connected component of this graph corresponds to a molecule, and the component itself is called *hydrogen-depleted molecular graph* (HDMG) of the molecule, since we generally do not care about hydrogen atoms.

The sensor has just detected some hexane molecules, which is a specific kind of hydrocarbons, consisting of 6 carbon atoms and 14 hydrogen atoms per molecule. There are five essentially different hexanes, also known as isomers:

Name	HDMG	Name	HDMG
n-hexane		2-methylpentane	
3-methylpentane		2,3-dimethylbutane	
2,2-dimethylbutane		-	-

Your task is to identify, for each hexane molecule detected, which specific type of hexane isomer it is. A molecule should be identified as any of the above isomers if its hydrogen-depleted molecular graph is isomorphic to the HDMG presented in the above table.

Input

The first line of input consists of only a single integer T ($1 \leq T \leq 200000$), denoting the number of hexane molecules detected.

Every five lines of the remaining part of the input specify a molecule detected. Each of the five lines contains two integers a, b ($1 \leq a, b \leq 6, a \neq b$), denoting a bond between carbon atoms a and b . In each molecule, the carbon atoms are numbered 1 through 6.

It is guaranteed that, for each molecule in the input, it is exactly one type of the hexane isomers listed above.

Output

For each molecule in the input, display the name of the isomer in one line.

Sample Input 1	Sample Output 1
2 1 2 2 3 3 4 4 5 5 6 1 4 2 3 3 4 4 5 5 6	n-hexane 3-methylpentane

Problem I

Travel Dream

Time limit: 3 seconds

Andy is an ordinary student at Nanjing University. One day, he found himself transmitted to a wonderland when he was dreaming. The wonderland consisted of several spots, with some bidirectional roads of various travel times connecting some pairs of them.

Attracted by the fascinating scenery, Andy wanted to visit some spots in the wonderland. He would like to select exactly k distinct spots and visit them one by one. After visiting the last spot, he must turn back to the first spot, because that was where his dream started. There must be a road between any two adjacent spots he selected, including the last one and the first one. Moreover, he wanted to maximize the total time spent in moving between the spots, so that he could better enjoy the beauties.

For example, in the first sample data, you may choose to start from spot 1, visit spots 3, 5, 2, and return to spot 1. The total time spent was 21 minutes, which turns out to be optimal.

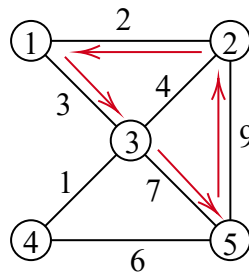


Figure 3: The first sample data.

However, finding such a traveling plan was too hard for Andy, so he wanted your help. Could you help him to find such an optimal plan?

Input

The first line of input consists of three integers n, m, k ($2 \leq n \leq 300, 1 \leq m \leq 300, 3 \leq k \leq 10$), denoting the number of spots and the number of roads in the wonderland, and the number of spots Andy would like to visit, respectively. The spots are numbered 1 through n .

The remaining m lines specify the roads between the spots. Each of these lines contains three integers u, v, t ($1 \leq u, v \leq n, u \neq v, 1 \leq t \leq 10^8$), representing a bidirectional road between spots u and v , and it took t minutes to travel along the road in either direction. It is guaranteed that there was at most one road between any pair of spots in the wonderland.

Output

Print the maximum total travel time in minutes as an integer in one line. If it is impossible to find any valid travel plan, output `impossible` instead.

Sample Input 1

```
5 7 4
1 2 2
1 3 3
2 3 4
4 3 1
5 3 7
4 5 6
2 5 9
```

Sample Output 1

```
21
```

Sample Input 2

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

Sample Output 2

```
impossible
```

Problem J

Wood Processing

Time limit: 3 seconds

In the wood industry, it is very common to make big wood boards by combining planks. To combine several planks into boards, the carpenter may cut some of the planks horizontally and discard one of the two parts, such that the heights of all planks are equal. Then, the planks are joined together, forming a big wood board. The height of the board is the common height of the planks, and the width of the board is the sum of the widths of the planks.

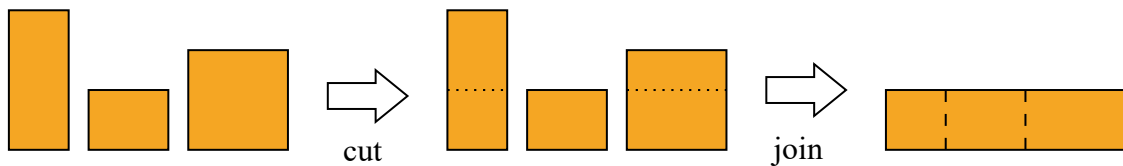


Figure 4: Making big boards from planks

However, cutting planks may result in huge wastes. The problem is, given n planks, determine the minimum total wasted area of planks to make k boards from these planks. You may freely reorder and combine these planks. Note that the mechanical properties of a plank are anisotropic, so you can't rotate the planks. Also, all planks must be used; you cannot discard any whole plank.

Input

The first line of the input contains two integers n, k ($1 \leq n \leq 5000, 1 \leq k \leq 2000, k \leq n$), denoting the number of planks given and the number of boards to make.

Each of the remaining n lines contains two integers w, h ($1 \leq w, h \leq 10^7$), denote the width and the height of a given plank.

Output

Output the minimum wasted area as an integer.

Sample Input 1

```
3 2
3 3
4 7
2 5
```

Sample Output 1

```
4
```

Sample Input 2

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

Sample Output 2

```
3
```