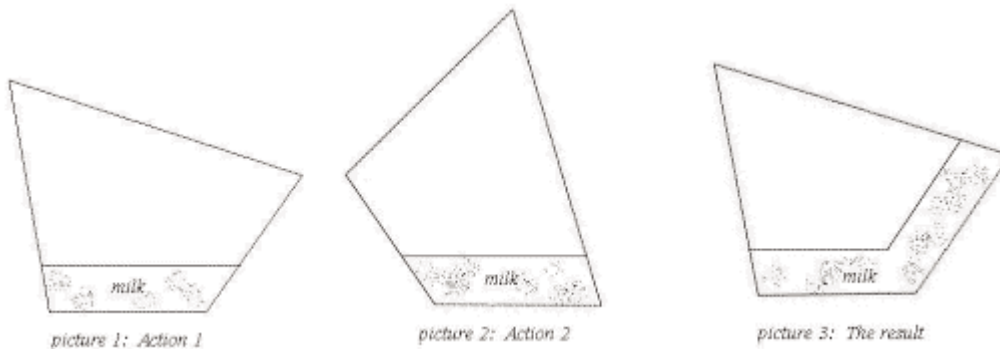


OIBH Online Programming Contest #1

Problem A

Nice Milk

Little Tomy likes to cover his bread with some milk. He does this by putting it in the cup so that one of its sides (called bottom side) touches the bottom of the cup, just as the picture below:



Since the milk in the cup is limited, only part of the bread is covered with milk(as shown in the pictures). That is, only the area between the surface of the milk and the bottom side of the bread is covered. Note that the distance between these two lines is always h - the depth of the milk, which is also known to him.

Tomy wants to cover this bread with largest possible area of milk in this way, but he doesn't want to do more than k actions. Help him, will you?

(You may assume that the cup is wide enough, wider than any side of the bread, so it's possible to cover any side completely)

Input

The input will contain no more than 10 test cases. Each test case begins with a line containing three integers n, k and h ($3 \leq n \leq 20$, $0 \leq k \leq 8$, $0 \leq h \leq 10$). A piece of bread is guaranteed to be a convex polygon of n vertices. In the following n lines, each line contains two integers x_i and y_i ($0 \leq x_i, y_i \leq 1000$), representing the Cartesianism coordinate of the i th vertex. The vertices are anti-clockwise numbered. The test case containing $n=0$, $k=0$, $h=0$ will terminate the input, you should not give an answer to this case.

Output

Output the area of the largest possible area of bread covered with milk with two decimal places. Output one line for each test case.

Sample Input

```
4 2 1
1 0
3 0
5 2
0 4
0 0 0
```

Sample Output

```
7.46
```

Problem B

Free Candies

Little Bob is playing a game. He wants to win some candies in it - as many as possible.

There are 4 piles, each pile contains N candies. Bob is given a basket which can hold at most 5 candies. Each time, he puts a candy at the top of one pile into the basket, and if there're two candies of the same color in it ,he can take both of them outside the basket and put them into his own pocket. When the basket is full and there are no two candies of the same color, the game ends. If the game is played perfectly, the game will end with no candies left in the piles.

For example, Bob may play this game like this (N=5):

Step1	Initial Piles		Step2	Take one from pile #2	
Piles	Basket	Pocket	Piles	Basket	Pocket
1234 1567 2333 4986 8721	nothing	nothing	1*34 1567 2333 4986 8721	2	nothing
Step3	Take one from pile #2		Step4	Take one from pile #3	
Piles	Basket	Pocket	Piles	Basket	Pocket
1*34 1*67 2333 4986 8721	2 5	nothing	1**4 1*67 2333 4986 8721	2 3 5	nothing
Step5	Take one from pile #2		Step6	Put two candies into his pocket	
Piles	Basket	Pocket	Piles	Basket	Pocket
1**4 1*67 2*33 4986 8721	2 3 3 5	nothing	1**4 1*67 2*33 4986 8721	2 5	A pair of 3

Note that different numbers indicate different colors, there are 20 kinds of colors numbered 1..20.

'Seems so hard...'Bob got very much puzzled. How many pairs of candies could he take home at most?

Input

The input will contain not more than 10 test cases. Each test case begins with a line containing a single integer n ($1 \leq n \leq 40$) representing the height of the piles. In the following n lines, each line contains four integers x_1, x_2, x_3, x_4 (in the range $1..20$). Each integer indicates the color of the corresponding candy. The test case containing $n=0$ will terminate the input, you should not give an answer to this case.

Output

Output the number of pairs of candies that the cleverest little child can take home. Print your answer in a single line for each test case.

Sample Input

```
5
1 2 3 4
1 5 6 7
2 3 3 3
4 9 8 6
8 7 2 1
1
1 2 3 4
3
1 2 3 4
5 6 7 8
1 2 3 4
0
```

Sample Output

```
8
0
3
```

Problem C

Farewell, my friend

Liz and Lilly used to be very good friends, but recently, they got into a silly quarrel and finally decided to say farewell to each other. 'I don't want to see you any more! I must put some new rocks somewhere so that no matter how I travel from my house, I can never see your face.' They both said.

They live in a small village which is divided into $n \times n$ grid. Liz always lives in the top-left corner (i.e grid (1,1)), and Lilly always lives in the bottom-right corner. (i.e grid(n,n)). Each grid of land is one of the following types: **land**, **lake** or **rock**. They cannot move across a **rock** or a **lake**, of course. And although people cannot see through a **rock**, it's easy for them to see through a piece of **land** or a **lake**. Note that they can only move north, south, east or west one grid at a time, NOT diagonally, and they are a bit shortsighted - they can only see things that are no more than k grids away **in front of** them (in the same row or column. They don't see anything diagonally).

Since they're both lazy, they want to put as few new rocks as possible. A new rock can only be put on a piece of land that at least one of the two girls can reach from her house. Note that they don't want to put new rocks too close to their houses, so the new rocks must be at least m grids away from both of the houses. By definition, grid(x_1, y_1) and grid(x_2, y_2) are supposed to be $\text{abs}(x_1 - x_2) + \text{abs}(y_1 - y_2)$ grids away from each other.

Input

The input will contain no more than 8 test cases. Each test case begins with a line containing three integers n , k and m ($5 \leq n \leq 20$, $1 \leq k \leq n$, $1 \leq m \leq n$) separated by a single space. The following n lines each contains n characters indicating the map of a village. The capital letter 'O' represents a **lake**, '*' represents a **rock**, '.' represents a piece of **land**. The test case containing $n=0$, $k=0$, $m=0$ will terminate the input, you should not give an answer to this case. No extra spaces at the beginning/end of each line.

Output

Output the least number of new rocks that must be put in order to separate them. Print your answer in a single line for each test case. If no solution found, you should output -1 in the corresponding line.

Sample Input

```
7 4 4
.....
.....*
....*O.
**.*.O.
```

```
...*...
.OO...*.
.....
0 0 0
```

Sample Output

2

Note

If they only set one new rock at (4,3), when Liz comes to (2,6) and Lilly comes to (5,6), they can still see each other. Thus, an additional rock at (2,6) must be put. The new map is shown below: ('N' represents a new rock)

```
.....
.....N*
.....*O.
**N*.O.
...*...
.OO...*.
.....
```

Problem D

Gift?!

There is a beautiful river in a small village. N rocks are arranged in a straight line numbered 1 to N from left bank to the right bank, as shown below.

[Left Bank] - [Rock1] - [Rock2] - [Rock3] - [Rock4] ... [Rock n] - [Right Bank]

The distance between two adjacent rocks is exactly 1 meter, while the distance between the left bank and rock 1 and the distance between Rock n and the right bank are also 1 meter.

Frog Frank was about to cross the river, his neighbor Frog Funny came to him and said,

'Hello, Frank. Happy Children's Day! I have a gift for you. See it? A little parcel on Rock 5.'

'Oh, that's great! Thank you! I'll get it.'

'Wait...This present is for smart frogs only. You can't get it by jumping to it directly.'

'Oh? Then what should I do?'

'Jump more times. Your first jump must be from the left bank to Rock 1, then, jump as many times as you like - no matter forward or backward, but your i th jump must cover $2*i-1$ meters. What's more, once you return to the left bank or reach the right bank, the game ends, and no more jumps are allowed.'

'Hmmm, not easy... let me have a think!' Answered Frog Frank, 'Should I give it a try?'

Input

The input will contain no more than 2000 test cases. Each test case contains a single line. It contains two positive integers N ($2 \leq N \leq 10^6$), and M ($2 \leq M \leq N$), M indicates the number of the rock on which the gift is located. A test case in which $N=0$, $M=0$ will terminate the input and should not be regarded as a test case.

Output

For each test case, output a single line containing `Let me try!` If it's possible to get to Rock m , otherwise, output a single line containing `Don't make fun of me!`

Sample Input

9 5
12 2
0 0

Sample Output

Don't make fun of me!
Let me try!

Note

In test case 2, Frank can reach the gift in this way:

Forward(to rock 4), Forward(to rock 9), Backward(to rock 2, got the gift!)

Note that if Frank jumps forward in his last jump, he will land on the right bank(assume that banks are large enough) and thus, lost the game.

Problem E

Legendary Pokemon

Do you know Pokemon? It's a kind of a very lovely cute sprite. In the world of pokemon, people live with their favorite pokemons and train them with love. One night, you saw a strange yet powerful pokemon in a narrow dark cave. 'It must be the legendary pokemon. If only I could catch him!' You thought. Setting out your best pokemon, you decided to catch it - the legendary pokemon.

As a skilled pokemon trainer, you know that all you have to do is to battle with it and throw a magic BALL to it during the battle. If you're lucky enough, the pokemon will be caught in the BALL, and become yours. It sounds easy, but the balls can not always catch pokemons, especially fierce ones, so you must be careful and follow a perfect strategy.

The battle is divided into rounds. In each round, your pokemon makes a move first, then does the wild pokemon (you can choose not to do anything). Each round, a pokemon makes a **move** or uses an **item** to attack its opponent or heal itself. You may also use a BALL to catch it in a round.

A move may be used as many times as you like, but can only be of the following three types:

1. Normal Attack

The move attacks the opponent. As a result, the HP(life value) of the opponent will decrease.

2. Status change

The move itself does not cause any damage, but you can make your opponent sleepy or confused. if your opponent's sleepy, he cannot make any move, if your opponent is confused, he will attack himself sometimes. These will be effective within 3 rounds (that is, the victim will be in the wrong status 3 times, and during that time, any Status-change type moves will not take effect).

3. Poison attack

This move attacks the opponent and makes him poisoned. If a pokemon is poisoned, it will lose an additional **poison value** units of HP BEFORE he can make a move. The effect lasts for 3 rounds(that is, the victim will be hurt by poison 3 times, during that time, any poison-type attack will still do damage, but will take no OTHER effect)

The damage that the opponent will take after being attacked by a Normal Attack or Poison attack is called **damage value** of the move; the probability that the move hits the target is called **accuracy** of that move. If a pokemon is confused by a move, the probability that it CHOOSE to attack itself is called **confusion value** of that move.

There is only one type of item available - **Hyper Potion**. It recovers a pokemon 200 HP.

A BALL could be of the following types:

1. **Poke Ball** - Used for catching pokemon. (BasicProbability=0.05)
2. **Great Ball** - Catches pokemon more easily than a pokeball. (BasicProbability=0.1)
3. **Ultra Ball** - Capture pokemon more easily than with the great ball. (BasicProbability=0.15)
4. **Level Ball** - If your pokemon is higher levels than the target pokemon then BasicProbability=0.18, otherwise, it is the same as a Great Ball.
5. **Love Ball** - Captures pokemon of an opposite gender than yours easily (BasicProbability=0.20), otherwise, BasicProbability=0.0

The probability that a BALL can catch a pokemon is computed this way:

$$P = \text{BasicProbability} + \text{CriticalLifeBonus} + \text{PoisonedBonus} + \text{StatusBonus}$$

CriticalLifeBonus=0.05 if and only if $50 < \text{the target's HP} \leq 100$.

CriticalLifeBonus=0.1 if and only if $0 < \text{the target's HP} \leq 50$.

PoisonedBonus=0.1 if and only if the target is poisoned.

StatusBonus=0.05 if and only if the target is in the wrong status(sleepy or confused).

All Bonus values are set to zero by default.

Initially, Both pockemons' HP are full(HP values can never be greater than their maximal values) and are in good health(not sleepy, confused or poisoned). Then, the battle starts. When one of the pokemon's HP is 0 or below 0, the battle ends immediately and the legendary pokemon goes away(you failed). When you catch it, the battle also ends(congratulations!)

Since the pokemon is wild, it has not been trained mentally. So it follows a very simple strategy during the battle:

1. In the i th round, he may decide to run away. The probability he makes this decision is $\text{run}[i]\%$; (he never fails to escape even if he's sleepy, confused or poisoned)
2. If he decided to continue the battle, he check if his HP. If it's not greater than 150, he uses a Hyper Potion if he has any(even if he's sleepy or confused), otherwise, he uses his only normal attack move - Cross Chop. Its Accuracy is 30%, but the damage value is 300.

It is well known that the legendary pokemon is at level 50 and is male. His initial HP is always 999, but people don't know how many Hyper Potions he has. People never saw him battle for 6 or more rounds, so you may assume this is also true for your battle.

Input

The input will contain no more than 20 test cases. Each test case begins with a line containing three integers l, g and HP describing your pokemon. l is your pokemon's level($1 \leq l \leq 100$), g ($0 \leq g \leq 1$) is its gender. 0 =male, 1 =female. HP is your initial(and maximal) HP value($1 \leq HP \leq 999$). The second line contains 5 integers. The i th integer is the value of $\text{run}[i]$, (i.e. $\text{run}[i]\%$ is the probability that the wild pokemon flee in his i th move.) It is guaranteed that $0 \leq \text{run}[i] \leq \text{run}[i+1]$ for all $1 \leq i \leq 4$, and that $\text{run}[5]=100$. The next line contains two integers $p1, p2$ ($0 \leq p1, p2 \leq 10$). $p1$ is the number of Hyper

Potions that you have, p2 is that of the wild pokemon's. The next line contains 5 integers b_1, b_2, b_3, b_4, b_5 ($0 \leq b_1, b_2, b_3, b_4, b_5 \leq 5$), the number of corresponding balls you have (i.e. b_1 for Poke Ball...). The next line contains a single integer k ($0 \leq k \leq 4$), the number of moves your pokemon masters. In the following k lines, each line begins with an integer t ($1 \leq t \leq 3$), representing the type of the move. If $t=1$, there are two integers following in the same line: damage value and $\text{accuracy} \times 100$; if $t=2$, there are two integers following in the same line: confusion value $\times 100$ and $\text{accuracy} \times 100$. Note that the move makes the opponent sleepy if and only if confusion value = 0. If $t=3$, there are three integers following in the same line: damage value, poison value and $\text{accuracy} \times 100$. All the values defined in the three types will be between 0 and 999 (inclusive) while $0 \leq \text{accuracy}$, $\text{confusion} \leq 100$. The test case containing $l=g=HP=0$ will terminate the input and should not be regarded as a test case.

Output

For each test case, output a single line containing the probability that you catch the pokemon if you follows a perfect strategy. Print your answer with four decimal places.

Sample Input

```
30 1 500
10 100 100 100 100
0 10
0 0 0 0 1
2
1 100 80
3 300 50 60
0 0 0
```

Sample Output

```
0.2340
```

Note

You should follow this strategy:

use move 2 (poison attack) in round 1, then throw his ball in round 2.

if this hits, the probability is $(1-10\%) \times (0.2+0.1) = 0.27$, otherwise, $\text{probability} = (1-10\%) \times 0.2 = 0.18$

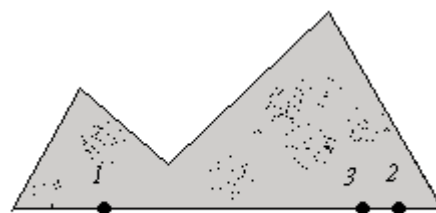
so the total probability = $60\% \times 0.27 + 40\% \times 0.18 = 0.2340$

(note that in some cases, the strategy is much more complex and may be dynamic - you may do different things in different conditions)

Problem F

Mysterious Mountain

A group of M people is chasing a very strange animal. They believe that it will stay on a mysterious mountain T , so they decided to climb on it and have a loot. The mountain looks ordinary, shown below:



Mountain T and 3 people

That is, the outline of the mountain consists of $N+1$ segments. The endpoints of them are numbered $0..N+1$ from left to right. That is to say, $x[i] < x[i+1]$ for all $0 \leq i \leq n$. And also, $y[0]=y[n+1]=0$, $1 \leq y[i] \leq 1000$ for all $1 \leq i \leq n$.

According to their experience, the animal is most likely to stay at one of the N endpoints numbered $1..N$. And... funny enough, they soon discover that $M=N$, so each of them can choose a different endpoint to seek for the animal.

Initially, they are all at the foot of the mountain. (i.e at $(s_i, 0)$) For every person i , he is planning to go left/right to some place $(x, 0)$ (where x is an integer - they do not want to take time to work out an accurate place) at the speed of w_i , then climb directly to the destination along a straight line (obviously, no part of the path that he follows can be OVER the mountain - they can't fly) at the speed of c_i . They don't want to miss it this time, so the teamleader wants the latest person to be as early as possible. How fast can this be done?

Input

The input will contain no more than 10 test cases. Each test case begins with a line containing a single integer N ($1 \leq N \leq 100$). In the following $N+2$ lines, each line contains two integers x_i and y_i ($0 \leq x_i, y_i \leq 1000$) indicating the coordinate of the i th endpoints. in the following N lines, each line contains three integers c_i, w_i and s_i describing a person ($1 \leq c_i < w_i \leq 100$, $0 \leq s_i \leq 1000$) - the climbing speed, walking speed and initial position. The test case containing $N=0$ will terminate the input and should not be regarded as a test case.

Output

For each test case, output a single line containing the least time that these people must take to complete the mission, print the answer with two decimal places.

Sample Input

```
3
0 0
3 4
6 1
12 6
16 0
2 4 4
8 10 15
4 25 14
0
```

Sample Output

```
1.43
```

Note

In this example, Person 1 goes to (5,0) and climbs to endpoint 2, Person 2 climbs directly to endpoint 3. person 3 goes to (4,0) and climbs to endpoint 1. Shown below:

