We are going to a funfair where there are n games $G_1, \ldots, G_n$. We want to play k games out of the n games, and we can choose the order in which we play them — note that we cannot play any game more than once. We have to specify these k games and their order before starting any game.

At each point in time, we have some amount of money, which we use in playing the games. At the beginning, we have x_0 Oshloobs of money. If before playing game G_i, we have x Oshloobs and we win in G_i, our money increases to $x + A_i$ for some $Ai \geq 0$. If we have x Oshloobs before playing game G_i and we lose in G_i, we lose L_i percent of x. The probability that we win game G_i (independently of other games) is P_i percents.

The goal is to play k of the games in such an order to maximize the expected amount of money we end up with after playing all k selected games in that order.

Input

There are multiple test cases in the input. The first line of each test case contains three space-separated integers n, k, and x_0 ($1 \leq k \leq n \leq 100$, $0 \leq x_0 \leq 10^6$). Each of the next n lines specifies the properties of game G_i with three space-separated integers A_i, L_i, and P_i ($0 \leq A_i, L_i, P_i \leq 100$). The input terminates with a line containing '`0 0 0`' which should not be processed.

Output

For each test case, output a single line containing the maximum expected amount of our final money rounded to exactly two digits after the decimal point.

Sample Input

```
2 2 100
10 0 50
100 10 20
2 1 100
10 0 50
100 10 20
0 0 0
```

Sample Output

```
117.00
112.00
```