There are some water, milk and wine in your kitchen. Your naughty little sister made some strange drinks by mixing them together, and then adds some sugar! She wants to know whether they taste good, but she doesn't want to try them herself. She needs your help.

Your sister knows that you don't want to drink them either (anyone wants to?), so she gives you a chance to escape: if you can guess the price of a special drink, she gives you freedom. Warning: she loves her special drink so much that you should never under-estimate its cost! That is, you're to find the most expensive possible price of it.

The price of each drink equals to its cost. If the amounts of water, milk, wine and sugar used in the drink are a_1, a_2, a_3 and a_4 respectively, and the unit costs of water, milk, wine and sugar are c_1, c_2, c_3 and c_4 respectively, then the drink costs $a_1c_1 + a_2c_2 + a_3c_3 + a_4c_4$. To give you some hope to win, she told you the costs of exactly n ordinary drinks. Furthermore, she promised that the total cost of sugar a_4c_4 is always a ***real number*** in the interval $[L, R]$, in any drink.

Sadly, you don't know the exact price of anything (you're a programmer, not a housewife!), but you know that water is the cheapest; wine is the most expensive, i.e., $0 \leq c_1 \leq c_2 \leq c_3$. Then the best thing you can do is to assume ***units costs can be any real numbers satisfying this inequality***.

Write a program to find the highest possible price of the special drink.

Input

The input contains several test cases. The first line of each test case contains three positive integers n, L, R ($1 \leq n \leq 100$, $0 \leq L \leq R \leq 100$). The next n lines each contain four non-negative integer a_1, a_2, a_3, p ($0 \leq a_1, a_2, a_3 \leq 100$, $0 \leq p \leq 10000$), the amount of water, milk and wine, and the price. The last line of the case contains three integers a_1, a_2, a_3 ($0 \leq a_1, a_2, a_3 \leq 100$), the drink to be estimated. The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the highest possible price to four decimal places. If the input is selfcontradictory, output '`Inconsistent data`'. If the price can be arbitrarily large, output '`Too expensive!`'.

Sample Input

```
1 3 5
1 2 3 10
2 4 6
1 2 4
1 1 1 1
1 1 1
1 3 8
0 1 0 17
0 0 1
3 1 2
2 1 3 14
1 5 1 15
7 3 2 21
4 1 6
2 0 2
45 31 53 4087
30 16 1 1251
11 51 34
0
```

Sample Output

```
Case 1: 19.0000
Case 2: Inconsistent data
Case 3: Too expensive!
Case 4: 26.2338
Case 5: 3440.3088
```