



3039 - LED Modding

North America - Southeast - 2004/2005

PC Modding has become the latest trendy hobby sweeping the nation. Since you don't have too much money to spare, and since modding is supposed to be a Do-It-Yourself sort of activity, you've decided to apply some simple modifications to your own PC.

One quick modification you figured wouldn't cost too much is to add some Light Emitting Diodes (LEDs) to your case. Thus, you bought yourself some LEDs, and, once they arrived to your house, you started looking for power. In order to draw power from one of the standard connectors on your motherboard you would need a connector. Alas, you forgot to order any of those. You're left with your back-up plan: take the power from the USB adapter available on your motherboard using a USB cable, and plug in your bright blue LEDs. You have the USB cable ready to go, but you realize that you still have a problem to solve: the USB port on the motherboard does not always provide power at the same voltage required by the LEDs. Looking around, you find a convenient box of resistors that might just do the trick.

Fortunately, you remember Ohm's Law:

$$\text{Voltage}(\text{involts}) = \text{Current}(\text{inamperes}) * \text{Resistance}(\text{inohms})$$

and realize you can use resistance to lower the voltage provided by the USB adapter to that expected by the LEDs. You even remember that if you can't reduce the voltage to exactly what they expect, you can still get a nice glow if you reduce it to below the expected voltage. Of course, if you provide too much voltage, you'll get a very short bright display, but you don't want to destroy your LEDs, so this is not acceptable. Likewise, if the voltage provided is 0 or lower, there will be no power to the LED, so this case is also unacceptable.

Input

The input file for this problem includes a series of 1 or more test cases.

Each test case begins with a line formatted as follows:

`< USB > < LED $_V$ > < LED $_mA$ > < N >`

Where:

`< USB >`

is a positive integer value indicating how many volts the USB adapter on the motherboard produces;

`< LED $_V$ >`

is a positive integer value indicating how many volts the LEDs expect to receive;

`< LED $_mA$ >`

is a positive integer value, in milliamperes (each milliamperes is a thousandth of an ampere) indicating the current at which the LEDs operate;

`< N >`

is a non-negative integer number indicating how many types of resistors are available in the test case.

This line is followed by $< N >$ lines each describing a type of resistor available. All types of resistors available in a given test case are different from each other. This means that their names and resistance values are all different. These lines are formatted as follows:

$< Resistor_Name > < Resistance >$

Where $< Resistor_Name >$ is a sequence of at most 40 alphabetic characters, and $< Resistance >$ is a positive integer, giving the resistance of that type of resistors, measured in ohms.

The end of input will be indicated by end of file.

Output

For each test case in the input file you should produce a single line in the output file formatted as follows:

$< Optimum_Res > < Resistor_Name >$

Where $< Optimum_Res >$ is the optimum resistance in ohms that should be added to the circuit to make the LEDs receive the power they expect, and $< Resistor_Name >$ is the name of the resistor, out of those available in the test case, that should be used. In displaying the optimal resistance, use 2 decimal digits, rounding any additional digits. When no resistor available provides the optimal resistance for the test case, you should select the resistor that is the closest to the optimum resistance and still makes the circuit work (providing to the LEDs less current than they expect).

If the circuit cannot be made to work with the given parts, instead of a line formatted as described above, your output file should include a line with only the ``IMPOSSIBLE'` string in it.

If the USB port provides acceptable voltage without addition to resistance to the LED, the output line should only include the ``--'` string.

Sample Input

```
5 3 20 2
Red 90
Orange 110
5 3 20 2
ThisOne 110
TheOtherOne 100
5 3 20 1
TooLittle 90
5 5 20 1
Sample 10
```

Sample Output

```
100.00 Orange
100.00 TheOtherOne
IMPOSSIBLE
---
```
