COM S 207 Bus

Write a program to simulate a bus operation. The bus departs from initial stop with certain amount of passengers. After that, the bus will stop three times before it hits final stop. For each stop, there will be some passengers getting on and off the bus. For every single passenger getting on the bus, a fixed amount of money will be charged.

You will need to create a bus object based on each set of input, which will be described in detail in following instructions.

Input

You will be inputting more than one set of FOUR lines

The first line contains a fixed route fare, and the number of passengers who get on the bus at the initial stop. For example: `1.50 4` means the bus fare is `$1.50` per passenger, and there are `4` passengers getting on the bus at the first stop.

After that, a bus stops at three bus stops before it hits the final stop. Each line of input represents one stop. The first number of input is the number of passengers who get on the bus, and the second one is the number of passengers who get off the bus. For example, `2 3` means at certain stop, `2` passengers get on the bus and `3` passengers get off the bus.

Important Note: We assume that passengers on board get off before others can get on, AND it's impossible for a bus to carry negative amount of passengers. Therefore, we set the following rule: whenever the number of passengers who get off the bus is greater than the number of passengers on board, we first ZERO OUT the passengers on bus, and then put others onto the bus.

To illustrate this, assume the bus arrives certain stop with 3 passengers, user inputs is `2 5`. Since `5 > 3`, we first zero out the passenger, and then add `2` to the busload, so finally the number of passengers on the bus will be 2.

Output

There will be two numbers in your output. The first one is the number of passengers who are going to the final stop; the second one is the total amount of money collected on this ride.

Sample Input

```
2.0 4
2 0
3 6
2 1
1.5 0
3 0
4 0
2 6
```

Sample Output

```
4 2.20000e+01
3 1.35000e+01
```

Now that you know what this program is to do, let us get into more details of what you need to do.

Bus class

1) The Bus class should have at least three instance variables:
   a) a double variable to indicate the fixed route fare of the bus
   b) a double variable to indicate total amount of money the bus collected on certain ride
   c) a int variable to indicate the number of passengers currently on the bus
2) the bus class should have at least three methods:
   a) **public** Bus(Double fare, int passenger) This is a constructor that sets the bus fare of the bus and the number of passengers at the initial stop.
   b) **public void** stop(int on, int off) This method will be called at each stop to get passengers on and off the bus. Please refer to Important Note in Sample Input section for rules.
   c) **public** String toString() that returns the output. Use the following statement to return value where pass and money are instance variables of the Bus class:

```
return String.format("%s %.5e",pass,money);
```

Main class

The Main class should have a main method (public static void main(String [] args) {})

The main method should keep reading lines

1. // process the FIRST line of the set
   a. Read the first line and create a bus object
2. // process the following three lines of the set
   a. Read each line; get passengers on and off the bus.
   b. Details are provided in Input section above.
3. // print the output

HOW TO CREATE A ZIP FILE USING ECLIPSE (FOR UPLOADING TO CYOJ)

In Eclipse, click on the folder that has your code. Then**, right click** will bring up options. Select EXPORT.

Then, select General, then select Archive File. Then, click next. That should bring up a screen similar to the below. Make sure all the options are selected as in the screen below. You will need to also specify the location of the archive file. Then click on finish and you should have a zip file that you can upload to cyoj site.