**AITP 2013 Java Contest**

**Before you Begin:**

**During the contest the only name you will have is Java###**

**(### is your assigned number)**

**PLEASE DO NOT PUT YOUR NAME ANYWHERE WITHIN THE FILES!**

* Do the problems in any order you like. They do not have to be done in orderfrom 1 to 4.
* The bonus points you find are nice to have but are not needed to have an advantage. I would not suggest to do them unless you have the time.
* Your program should not print extraneous output. Follow the instruction as given in the problem.
* **Start your program with the given template**. The variable values that are given in the template will be changed by the judge to test your program.
* Please name the program the name that is given for each program.

**(90 Total Points)**

**Problem 1: MEAN/MEDIAN (20 Points)**

**Java Program Name: meanMedian.java**

**Problem 1**

The **mean** of a set of numbers is the sum of all the numbers of a list divided by the number of numbers in the list. For example {1, 1, 1, 2, 5, 8} has the mean *18/6 =* **3**.

The **median** of a set of numbers is the average of the number(s) that appears in the middle of the set, if the numbers are sorted from the lowest to the highest. If there is an even number of numbers in the list, then the median is the middle two numbers.

Your program is going to be given an array of numbers:

{ 11, 23, 27, 31, 35, 46, 51, 74}

If there is a nonzero decimal, please show up to two decimal places.

**Sample Inputs Sample output data**

Type: Array Type: String

1, 1, 1, 2, 5, 8 The mean is 3 and the median is 1.5.

2, 2, 2, 5, 8 The mean is 3.8 and the median is 2**.**

2, 4, 4, 4, 5, 8, 12 The mean is 5.57 and the median is 4.

**Template (Start with this code)**

|  |
| --- |
| **public** **class** meanMedian { **public** **static** **void** main(String[] args) { **int[] numArray = { 11, 23, 27, 31, 35, 46, 51, 74};** // **TODO**:  }} |

Your program is going to output the following:

**EXAMPLE OUTPUT WITH EXAMPLE DATA (**2, 4, 4, 4, 5, 8, 12)

The MEAN is: 5.57!

The MEDIAN is: 4!

**Problem 2: Odd/Even Product (20 Points)**

**Java Program Name: oddEvenProduct.java**

Write a program that calculates the **product** of the **odd integers** and the **product** of the **even integers** from 1 up to (and including) N.

For instance: if N = 7, the product of all even numbers is 48 and the product of all odd integers is 105.

**Sample Inputs Sample output data**

N = 7

Odd 1 x 3 x 5 x 7 = 105

Even 2 x 4 x 6 = 48

**Template (Start with this code)**

|  |
| --- |
| **public** **class** oddEvenProduct { **public** **static** **void** main(String[] args) { **int** N = 20; // **TODO**:  }} |

**Output Example With Example Data (N = 7)**

The product of all odd numbers is 105!

The product of all even numbers is 48!

**Problem 3: Arts and Cats (20 Points)**

**Java Program Name: meow.java**

Draw a cat to the console:

                     \_
                    / )
                   ( (
     ^.\_.^  . """". ) )
    / , , \/      \/ /
   =\  t  /=    /   /
     `---`  """|   /
         || |  \\ \
  MEOW   ((,\_|  ((,\_\

**Bonus Points! (Do this if ONLY if you have finished part 4 and you have time)**

After you have finished the cat, draw a Halloween themed ASCII art.

Draw the jack o’ lantern below or feel free to get some art from anywhere.

**,\_   .-.**

**\ `\/ '`**

**\_.--"|  |"--.\_**

**.' '  '`--`'  ' '.**

**/  '  /\    /\  '  \**

**;  '  /o\_\  /o\_\  '  ;**

**|  .    . /\ :    .  |**

**;  . /\ .'--'. /\ .  ;**

**\  .\ \/\/\/\/ /.  /**

**'.\_:\\_/\\_\_/\\_/.\_.'**

**`'--'--'--'`**

**Problem 4 (30 Points)**

You just got hired at Snowstorm entertainment as an RPG developer. Your first task that the lead developer has assigned to you is to take a **combat damage algorithm** (provided below) and programmatically apply it to the game to calculate damage in a test environment. You are given the following test variables:

**Test DATA**

A = Attack Stat //Will always be between 15-100

D = Defense Stat //Will always be between 15-100

L = Attackers Level //Will always be between 1-100

C = Critical Hit //Will be initialized to 1

**Combat Damage Algorithm**

 ((((L \* 0.4) \* C) \* A \* 40 / 50 / D) + 2)

**Template (Start with this code)**

|  |
| --- |
| **public** **class** damageCalculator { **public** **static** **void** main(String[] args) { **int attackStat = 80;** **int defenseStat = 56;** **int attackersLevel = 70;** **int defendersHealth = 60;**  **int totalDamage;**  **int critical = 1;**  // **TODO**:  }} |

Using the data in the template as test inputs, you are the take this data and programmatically calculate the damage done, **round the damage to the nearest whole number,** and print out how much damage is done as well as how much health the defenders has left. **If the defenders health falls below zero you need to output that the defender has lost the battle.**

(Hint: Be sure to uses this test data to test your code. I would do the rounding feature last)

**Output Example With Example Data**

**Example DATA**

Attack Stat = 100

Defense Stat = 40

Attackers Level = 50

Defenders health = 100

Critical Hit = 1

The attacker hits for 42 damage!

The current health of the defender is 58.

**Output Example With Example Data**

Attack Stat = 90

Defense Stat = 30

Attackers Level = 100

Defenders health = 50

Critical Hit = 1

The attacker hits for 98 damage!

The defenders' health has fallen to zero. The defender has lost the battle!

**Bonus Points:** There is a 15% chance to critically hit for **double** the damage. Can you program this random chance and apply it into the equation? (**Hint:** all you need to do is set critical to 2, only 15% of the time)

**The output for the damage will look like this.**

The attacker hits for 194 damage!(CRITICAL!)

**When You Are Done:**

Please put all of your files into a folder called Java### and call the contest coordinator over to collect the folder.