**SUTD 50.001 Introduction to Information Systems and Programming**

**Problem Set 1X**

|  |
| --- |
| **Note:**   * **For all questions, please access the vocareum link found at eDimension for the starter code and to submit.** * **The Vocareum link is for submission only. Please work on the problems in Android studio, and this includes writing code for the test cases.** * **To prevent hard-coding, test cases used in Vocareum *may* be different from those provided here and will not be given to you.** * **Tips on using Android Studio are found at the end of this problem set** |

**Week 1**

**Question 1**

**Title: Adding Time**

[15 points] Write a function that takes two positive hour and minute values, adds them up and returns a string describing the result, in this format:

Eg. For *hour1* = 1, *min1* = 30, *hour2* = 1, *min2* = 31

addTime( 1,30, 1, 31)

Expected output:

3 hours 1 minutes

Note the words "hours" and "minutes" do not need to be adjusted for singular/plural

Assume:

* *min1*, *min2* are values between 0 and 59, both values inclusive
* *hour1*, *hour2* are values between 0 and 9, both values inclusive

**Test Case: Expected Output:**

addTime(0, 20, 0, 30) 0 hours 50 minutes

addTime(0, 40, 0, 30) 1 hours 10 minutes

addTime(1, 30, 1, 31) 3 hours 1 minutes

addTime(6, 30, 4, 45) 11 hours 15 minutes

**Question 2a**

**Title: Sum up to a given number**

[10 points] Write a function that takes in positive integer *n* and returns the sum of

1 + 2 + 3 + ... + n - 1 + n

Output is type int.

Assume: *n* is greater than 0

You may use a for-loop or the sum of series formula

(Test case inputs: 9 Expected output: 45)

**Test Case:**

sumUpTo(9);

**Expected output:**

45

**Question 2b**

**Title: Sum all elements in Array**

[10 points] Suppose that *array* is a variable of type int[]. Write a function that returns the sum of all elements in *array*.

Output is type int.

Assume: Integer overflow will not happen, and *array* will have zero or more elements.

(Test case inputs: -12, 3, 20, 21, -15, 2 Expected output: 19)

**Test Case:**

int a[] = {-12, 3, 20, 21, -15, 2};

int b[] = {};

int c[] = new int[10];

sumIntArrayAll(a);  
sumIntArrayAll(b);  
sumIntArrayAll(c);

**Expected output:**

19  
0  
0

**Question 2c**

**Title: Sum all elements in Array that are larger than 20**

[X points] Suppose that *array* is a variable of type int[]. Write a function that returns the sum of all elements in *array* that are larger than 20.

Output is type int.

Assume: Integer overflow will not happen. *array* can have zero elements or more.

(Test case inputs: -12, 3, 20, 21, -15, 22 Expected output: 43)

**Test Case:**

int a[] = {-12, 3, 20, 21, -15, 22};

int b[] = {};

sumIntArrayTwenty(a)  
sumIntArrayTwenty(b)

**Expected output:**

43

0

**Question 2d**

**Title: Count all even numbers in Array**

[10 points] Suppose that *array* is a variable of type int[]. Write a function that returns the count of the number of even numbers (which could be positive or negative) stored in *array*. Output is type int.

Assume:

* Integer overflow will not happen
* 0 is considered an even number
* *array* will have zero elements or more

(Test case inputs: -12, 3, 0, 21, -15, 2 Expected output: 3)

**Test Case (0 is considered even number):**

int a[] = {-12, 3, 0, 21, -15, 2};

int b[] = {};

countEvenNumbers(a);  
countEvenNumbers(b);

**Expected output:**

3  
0

**Test Case (**array only contains positive integers**):**

int a[] = {2, 3, 9, 5, 4, 1, 1}

countEvenNumbers(a)

**Expected output:**

2

**Question 3**

**Title: Number of terms required**

[10 points] The sum of series 1/1^2 + 1/2^2 + 1/3^2 + … + 1/n^2 converges to pi^2/6:

Write a function that takes in *p*,a double that can be assumed to be between 0 and 0.999, and returns the number of terms *N* needed in this series to achieve a sum that is equal or larger than a fraction *p* of pi^2/6:

Find *N* for:

Output is type int.

Assume: *p* is between 0 and 0.999

(Test case inputs: 0.9 Expected output: 6)

Hints.   
The constant pi in Java can be accessed with Math.PI. Use a while loop.

**Test Case:**

termsRequired(0.9)

**Expected output:**

6

**Question 4**

**Title: Binary to Decimal**

[10 points] Write a function that:

* takes in a String *s* that contains only the characters ‘0’ and ‘1’.
* *s* represents a **positive** binary number
* this function returns the decimal equivalent of this binary number as an integer value

Output is type int.

Assume: Integer overflow will not happen when calculating the result. *s* has at least one element and will only contain 1 or 0.

(Test case inputs: “1011” Expected output: 11)

**Background Information.**

1011 (base 2) = 1×23 + 0×22 + 1×21 + 1×20 = 11 (base 10)

Pseudocode (Loop solution)

1. Given a string s of length n with characters 1 and 0 only
2. Initialize total to 0
3. For index i from n -1 to 0
   1. Get the character at position i and determine its equivalent int value
   2. Calculate the power of 2 for that position (e.g. power of 2 at position n-1 is 0)
   3. Calculate the equivalent decimal value and add it to the total.

Note. If you can, explore more than one way to solve this.

**Test Case:**

binaryToDecimal(“1011”)   
binaryToDecimal(“0”)

**Expected output:**

11  
0

**Android Studio Tips**

1. To run a pure Java project in android studio

* Launch Android Studio
* Select **Start A New Android Studio Project**
* Name your project at the **Create Android Project**, click Next,
* Select all defaults at **Target Android Devices** (i.e. just click Next)
* At **Add An Activity to Mobile**, select **Add No Activity**, click Finish
* Wait for a while …
* Select **File** à **New à New Module** and choose **Java Library**, enter the relevant information and click Finish.

For more information:

https://stackoverflow.com/questions/16626810/can-android-studio-be-used-to-run-standard-java-projects

1. Automatically generate constructors, getters and setters:

* Place the cursor anywhere within your class definition.
* Select **Code** à **Generate** and choose the relevant option.

1. To pass arguments to your public static void main() function:

* Select **Run** à **Edit Configurations**
* Enter the arguments in the **Program Arguments** box