**Task 1**

Write a Java method that takes a string as an argument. Your task is to calculate the number of uppercase letters and lowercase letters and print them in the method.

===================================

**Method Call:**  
upper\_lower\_finder(“The quick Sand Man”)  
**Output:**  
No. of Uppercase characters : 3  
No. of Lowercase Characters: 12  
============================  
**Method Call:**  
upper\_lower\_finder(“HaRRy PotteR”)  
**Output:**  
No. of Uppercase characters : 5  
No. of Lowercase Characters: 6

**Task 2**

Write a method called **foo\_moo** that takes a number as an argument and **returns** the following statements according to the below-mentioned conditions. Then, finally prints the statement in the method call.

* If the number is divisible by 2, it should return "Foo".
* If the number is divisible by 3, it should return "Moo".
* If the number is divisible by both 2 and 3, it should return "FooMoo".
* Otherwise, it returns "Boo".

===================================

**Example1:**   
**Method Call:**  
foo\_moo(5)  
**Output:**  
Boo

**Example2:**   
**Method Call:**  
foo\_moo(4)  
**Output:**  
Foo

**Example3:**   
**Method Call:**  
foo\_moo(6)  
**Output:**  
FooMoo

**Task 3**

Write a method called **calculate\_tax** that takes 3 arguments: your age, salary, and current job designation.

Your first task is to take these arguments as **user input** and pass these values to the method.

Your second task is to implement the method and calculate the tax as the following conditions:

* **NO TAX IF YOU ARE LESS THAN 18 YEARS OLD.**
* **NO TAX IF YOU ARE THE PRESIDENT OF THE COMPANY**
* No tax if you get paid less than 10,000
* 5% tax if you get paid between 10K and 20K
* 10% tax if you get paid more than 20K

Finally return this tax value. Then print the returned value in the method call.

===================================

**Hints:**  
Here the job designation is a string, so it can be written in both uppercase and lower cases. So, you need to check the value ignoring the case.

===================================

**Example1:**   
**Input:**  
16  
20000  
Student  
**Method Call:**  
calculate\_tax(16, 20000, “Student”)  
**Output:**  
0

===================================

**Example2:**   
**Input:**  
20  
18000  
assistant manager  
**Method Call:**  
calculate\_tax(20, 18000, “assistant manager”)  
**Output:**  
900.0

===================================

**Example3:**   
**Input:**  
20  
122000  
president  
**Method Call:**  
calculate\_tax(20, 122000, “president”)  
**Output:**  
0

**Task 4**

Write a method named **year\_month\_day\_finder** which will take 1 argument, number of days.

Your first task is to take the number of days as user input and pass the value to the function.

Your second task is to implement the method and calculate the total number of years, number of months, and the remaining number of days as output. No need to return any value, print inside the method.

**Note:** Assume, each year to be 365 days and month to be 30 days.

**Example01**

**Input:**  
4330  
**Method Call:**  
year\_month\_day\_finder (4330)  
**Output:**  
11 years, 10 months and 15 days

================================

**Example02**

**Input:**  
2250  
**Method Call:**  
year\_month\_day\_finder (2250)  
**Output:**  
6 years, 2 months and 0 days

**Task 5**

Write a method called **make\_square** that takes two integer numbers in the parameter as a range of numbers (starting point and ending point (included)). The method should **return an array** that contains the squares of those numbers.

**[Hint: The return type of the method will be int [ ] ]**

=====================================================

**Example1:**   
**Method Call:**  
make\_square(1,3)  
**Output:**  
1 4 9

====================================

**Example2:**   
**Method Call:**  
make\_square(5,9)  
**Output:**  
25 36 49 64 81

**Task 6**

Write a method called **count\_duplicate** that takes an integer array in the parameter and **returns the total number of** duplicate values. Finally, print the number outside the method. =====================================================

**Example1:**   
**Method Call:**

int arr [] = {9, -5, 7, 9, -5, 5, 7};  
int duplicates = count\_duplicate(arr)  
**Output:**

3

**Example2:**   
**Method Call:**

int arr [] = {11, -5, 11, 55, 5, 55, 2};  
int duplicates = count\_duplicate(arr)  
**Output:**

2

**Task 7**

Write a method called **even\_array** that takes an integer array in the parameter and **returns a new array** that contains only the even values. Then print the returned array after the method call.

**[Hint: The return type and the parameter type of the method will be int [ ] ]**

=====================================================

**Example1:**   
**Method Call:**

int arr [] = {92, 10, 7, 8, 56, 5, 7};  
even\_array (arr)  
**Output:**  
92 10 8 56

**Task 8**

Write a method which will take 2 arguments. They are:

* Sentence
* Position

Your first task is to take these arguments as user input and pass these values to the method parameters.

Your second task is to implement the method and remove the characters at the index number which is divisible by the position (Avoid the index number 0 as it will always be divisible by the position, so no need to remove the index 0 character). Finally, add the removed characters at the end of the new string.

Return the value and then finally, print the new string at the method call.

**Input:**  
"I love programming."  
3  
**Method call:**  
function\_name("I love programming.", 3)  
**Output:**  
I lveprgrmmngo oai.

==============================================

**Input:**  
"Python is easy to learn. I love python."  
6  
**Method call:**  
function\_name("Python is easy to learn. I love python.", 6)  
**Output:**  
Pythonis eay to earn.I lov pythn. sl eo

**Task 9**

Write a method that will take 4 arguments. They are:

* starting value(inclusive)
* ending value(exclusive)
* first divisor
* second divisor

Your first task is to take these arguments as user input and pass these values to the method call.

Your second task is to implement the method and find all the numbers that are divisible by the **first divisor or second divisor but not both** from the starting value(inclusive) and ending value(exclusive). Add all the numbers that are divisible and finally return this value. Print the returned value in the method call.

=====================================================

**Input:**  
10  
40  
4  
7  
**Function Call:**  
method\_name(10, 40, 4, 7)  
**Output:**  
210  
================================  
**Input:**  
5  
100  
3  
4  
**Function Call:**  
method\_name(5, 100, 3, 4)  
**Output:**  
2012

**Not in the syllabus**

**Task 10**

Write a Java method called **value\_Add** that will take some arguments, add the values, and print the result outside the method.

[Hint: Check the method call and think of changing the signature of the method to handle different arguments]

**Method Call1:**

value\_Add (4, 5)

**Output**

9

**Method Call2:**

value\_Add (“Hello”, “World”)

**Output**

HelloWorld

**Method Call3:**

value\_Add (11, 5.9) //Consider 5.9 as double

**Output**

16.9

**Method Call4:**

value\_Add (11, 34, 90)

**Output**

135

**Task 11**

Write a recursive method called **string\_rev** that takes a string and its length as arguments and reverses the string.

**Method Call1:**

string\_rev(”ABCDEF”, 6)

**Output**

FEDCBA

**Method Call2:**

string\_rev(”Humpty Dumpty”, 13)

**Output**

ytpmuD ytpmuH

**Task 12**

Write a recursive method called **number\_print** that takes two numbers (Starting and ending points) as arguments and prints all the numbers from the starting point to the ending point.

**Method Call1:**

number\_print(20, 30)

**Output**

20 21 22 23 24 25 26 27 28 29 30

**Method Call2:**

number\_print(10, 30)

**Output**

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

**Task 13**

Write a recursive method to print the sum of the first N numbers.

**Method Call:**

sumOfNumbers(4)

**Output:**

10

**Explanation:**

N = 4 so 1+2+3+4 = 10

**Task 14**

Write a recursive method to print the Factorial N number.

**Method Call:**

recursiveFact(4)

**Output:**

24

**Task 15**

Write a recursive method to calculate the power of a number

**Method Call1:**

recursivePow(2, 3)

**Output:**

8

**Method Call2:**

recursivePow(4, 2)

**Output:**

16

**Advanced Task**

Run the code below in Dr. Java and check the outputs. For the array, why do arr1 and new\_arr have the same values? Hint: Print the location of both the arrays.

**import java.util.Arrays;**

**class A{**

**public static void main (String args[]){**

**int arr1 [] = {4,5,6};**

**String str1 = "Hello";**

**int num1 = 34;**

**//Passing the String**

**String new\_str1 = data\_pass(str1);**

**System.out.println("Values after the method call");**

**System.out.println(new\_str1);**

**System.out.println(str1);**

**//Passing the integer**

**int new\_num1 = data\_pass(num1);**

**System.out.println("Values after the method call");**

**System.out.println(new\_num1);**

**System.out.println(num1);**

**//Passing the array**

**int new\_arr [] = data\_pass(arr1);**

**System.out.println("Values after the method call");**

**System.out.println(Arrays.toString(new\_arr));**

**System.out.println(Arrays.toString(arr1));**

**}**

**public static int [] data\_pass(int [] a){**

**a[1] = 999;**

**return a;**

**}**

**public static String data\_pass(String a){**

**a+="999";**

**return a;**

**}**

**public static int data\_pass(int a){**

**a+=999;**

**return a;**

**}**

**}**