1.SumOdd

Write a program to read an integer and find the sum of all odd numbers

from 1 to the given number. [inclusive of the given number]

if N = 9 [ 1,3,5,7,9]. Sum = 25

Include a class UserMainCode with a static method addOddNumbers which

accepts the number n. The return type is the integer based on the problem

statement.

Create a Class Main which would be used to accept the integer and call

the static method present in UserMainCode.

Input and Output Format:

Input consists of a integer.

Output consists of a integer.

Refer sample output for formatting specifications.

Sample Input 1:

6

Sample Output 1:

9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.Sum of cubes and squares of elements in an array

Write a program to get an int array as input and identify even and odd

numbers. If number is odd get cube of it, if number is even get square of

it. Finally add all cubes and squares together and return it as output.

Include a class UserMainCode with a static method addEvenOdd which

accepts integer array as input.

The return type of the output is an integer which is the sum of cubes and

squares of elements in the array.

Create a class Main which would get the input and call the static method

addEvenOdd present in the UserMainCode.

Input and Output Format:

Input consists of integer array.

Output is an integer sum.

Refer sample output for formatting specifications.

Sample Input 1:

5

2

6

3

4

5

Sample Output 1:

208

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3.Removing vowels from String

Given a method with string input. Write code to remove vowels from even

position in the string.

Include a class UserMainCode with a static method removeEvenVowels which

accepts a string as input.

The return type of the output is string after removing all the vowels.

Create a Main class which gets string as an input and call the static

method removeEvenVowels present in the UserMainCode.

Input and Output Format:

Input is a string .

Output is a string .

Assume the first character is at position 1 in the given string.

Sample Input 1:

commitment

Sample Output 1:

cmmitmnt

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.Largest Chunk

Write a program to read a string and return the length of the largest

"chunk" in the string.

A chunk is a repetition of same character 2 or more number of times. If

the given string doest not contain any repeated chunk of characters

return -1.

Include a class UserMainCode with a static method getLargestSpan which

accepts the string. The return type is the integer.

Create a Class Main which would be used to accept the string and call

the static method present in UserMainCode.

Input and Output Format:

Input consists of a string.

Output consists of integer.

Refer sample output for formatting specifications.

Sample Input 1:

This place is soooo good

Sample Output 1:

4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.Digits - II

Write a program to read a non-negative integer n, compute the sum of its

digits. If sum is greater than 9 repeat the process and calculate the sum

once again until the final sum comes to single digit.Return the single

digit.

Include a class UserMainCode with a static method getDigitSum which

accepts the integer value. The return type is integer.

Create a Class Main which would be used to accept the string and call

the static method present in UserMainCode.

Input and Output Format:

Input consists of a integer.

Output consists of integer.

Refer sample output for formatting specifications.

Sample Input 1:

9999

Sample Output 1:

9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6.String Splitter

Write a program which would accept a string and a character as a

delimiter. Apply the below rules

1. Using the delimiter, split the string and store these elements in

array.

2. Reverse each element of the string and convert it into lowercase.

Include a class UserMainCode with a static method manipulateLiteral which

accepts the string and character. The return type is the string array

formed.

Create a Class Main which would be used to accept the string and

characterand call the static method present in UserMainCode.

Input and Output Format:

Input consists of a string and character.

Output consists of a string array.

Refer sample output for formatting specifications.

Sample Input 1:

AAA/bba/ccc/DDD

/

Sample Output 1:

aaa

abb

ccc

ddd

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7.Playing with String - I

Given a string array and non negative integer (n) apply the following

rules.

1. Pick nth character from each String element in the String array and

form a new String.

2. If nth character not available in a particular String in the array

consider $ as the character.

3. Return the newly formed string.

Include a class UserMainCode with a static method formString which

accepts the string and integer. The return type is the string formed

based on rules.

Create a Class Main which would be used to accept the string and integer

and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a an integer which denotes the size of the array

followed by the array of strings and an integer (n).

Output consists of a string .

Refer sample output for formatting specifications.

Sample Input 1:

4

ABC

XYZ

EFG

MN

3

Sample Output 1:

CZG$

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_