**Assignments on Fundamentals Of Programming**

1. Define a method which returns the sum of digits of the given two digit number.

Write the method with the following specifications:

Name of method **getSumOfDigits()** // which accepts an integer value as argument and return the sum of it's digits.  
Arguments: one argument of type integer  
Return Type: an integer value  
  
Specifications: The value returned by the method *getSumOfdigits()* is determined by the following rules:

* if the given value is in between 10 and 99, return sum of it's digits. Example: if x = 34, return 7
* if the given value is negative, return -3
* if the given value is greater than 99, return -2
* if the given value is in between 0 and 9, return -1

1. Define a method which returns the difference of digits of the given two digit number.

**Note:** You should substract the units position value from tens positon value, the return value may be negative.

Write the method with the following specifications

Name of method ***getDiffOfDigits()*** // which accepts an integer value as argument and return the difference of it's digits.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***getDiffOfdigits()*** is determined by the following rules

if the given value is in between 10 and 99, return difference of it's digits.

**Example:** if x = 83,  8 - 3 return 5. if x = 38, 3 - 8 return -5.

if the given value is negative, return -3

if the given value is greater than 99, return -2

if the given value is in between 0 and 9, return -1

1. Define a method which returns the next multiple of 100 for the given number.

Write the method with the following specifications

Name of method ***getNextMultipleOf100()*** // which accepts an integer value as argument and return the next multiple of 100.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***getNextMultipleOf100()*** is determined by the following rules

if the given value is negative or zero, return -1

if the given value is positive, return the next multiple of the given number.

Example: if x = 123, return 200.

1. Define a method which returns the 1 if the given three digit number is palindrome, in other case return 0.

Write the method with the following specifications

Name of method ***isPalindrome()***  // which accepts an integer value as argument and return true if the given number is palindrome, else retrun false.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method***isPalindrome()*** is determined by the following rules

if the given number is an three digit number, retun 1 if the number is palindrome, else return 0.

**Example:** if x = 232, return 1. if x = 345, return 0

if the given number is negative or zero, return -1

if the given number is not an three digit number, return -2

1. Define a method which returns the 1 if the given number is even, in other case return 0

Write the method with the following specifications

Name of method ***isEven()*** // which accepts an integer value as argument and return 1 if the given number is even, else retrun 0.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method **isEven()** is determined by the following rules

if the given number is an even number, return 1 else return 0. Example if x = 22, return 1. if x = 35, return 0

if the given number is negative or zero, return -1

1. Define a method which returns the greatest number among two numbers.

Write the method with the following specifications

Name of method ***getGreatest()***  / / which accepts two integer values as argument and return the greatest value.

Arguments: two argument of type integer

Return Type: an integer value

Specifications: The value returned by the method***getGreatest()*** is determined by the following rules

if any of the given numbers are negative, return -1.

if any of the given numbers are zero, return -2.

if the given numbers are positive, return the greatest.

1. Define a method which returns the least number among two numbers.

Write the method with the following specifications

Name of method ***getLeastNum()*** // which accepts two integer values as argument and return the least value.

Arguments: two argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***getLeastNum()*** is determined by the following rules

if any of the given numbers are negative, return -1.

if any of the given numbers are zero, return -2.

if the given numbers are positive, return the least number.

1. Define a method which returns the number itself if it is an even number, if the number is odd then return the next multiple of 10.

Write the method with the following specifications

Name of method ***oddRounder()*** // which accepts an integer value as argument and return the same value if it is an even number, if the value is odd then return the next multiple of 10. Example if x = 24 then return 24, if x = 25 then return 30.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***oddRounder()*** is determined by the following rules

if any of the given number is negative, return -1.

if any of the given number is zero, return -2.

if the given number is even, return the same number.

if the given number is odd, return the next multiple of 10.

1. Define a method which returns the 1 if the given number is positive, return -1 if the given number is negative, return 0 if the given number is 0.

Write the method with the following specifications

Name of method***findSign()*** // which accepts an integer value as argument and return 1 if the argument value is positive, return -1 in case of negative value, return 0 if the argument value is 0.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***findSign()*** is determined by the following rules:

if any of the given number is positive, return 1.

if any of the given number is negative, return -1.

if any of the given number is zero, return 0.

1. Define a method which returns the string as "Even" if the given number is an even number, return "Odd" if the given number is an odd number, return string as "Invalid Input" if the given number is lessthan or equal to 0.

Write the method with the following specifications

Name of method ***isEvenOrOdd()*** // which accepts an integer value as argument.

Arguments: one argument of type integer

Return Type: an String value (Even/Odd/Invalid Input)

Specifications: The value returned by the method ***isEvenOrOdd()*** is determined by the following rules

if the given number is negative or zero, return "Invalid Input"

if the given number is even, return "Even"

if the given number is odd, return "Odd"

1. Write a method which returns the square of the given number if it is an even, return cube of the given number if it is an odd number.

Write the method with the following specifications

Name of method ***calculate()*** // which accepts an integer value as argument and return square of the given value if it is an even, return cube of the given value if it is an odd number.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***calculate()*** is determined by the following rules

if the given number is negative or zero, return -1.

if the given number is even, return square of the number.

if the given number is odd, return cube of the given number.

1. Write a method which returns the sum of three numbers after rounding off each number to the next multiple of 10. If any of the given number is multiple of 10 don’t change it’s value.

For Example:   
if value of X is 56 round it's value to 60  
if value of X is 30, don't change it's value.

Write the method with the following specifications

Name of method ***sumOfMultiples()*** // which accepts three integer value as argument and return the sum of three numbers after rounding off each number to the next multiple of 10.

Arguments: three argument of type integer

Return Type: an integer value

Specifications: The value returned by the method **sumOfMultiples()**is determined by the following rules

if any of the given number is negative or zero, return -1.

in other case return the sum of all three rounded values.

**Example**

if a = 23, b = 34, c = 69

   30 + 40 + 70 = 140

if a = 23, b = 34, c = 50

   30 + 40 + 50 = 120

1. Write a method which returns the sum of three rounded numbers. If the right most digit of the number is lessthan 5, then round off it's value to the previous multiple of 10 otherwise if the right most digit of the number is greater or equal to 5, then round off to the next multiple of 10.

Write the method with the following specifications

Name of method***sumOfRoundedValues()*** // which accepts three integer value as argument and return the sum of three rounded numbers.

**Example**

if a = 23, b = 34, c = 66

   20 + 30 + 70 = 120

if a = 23, b = 37, c = 55

   20 + 40 + 60 = 120

Arguments: three argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***sumOfRoundedValues()*** is determined by the following rules

if any of the given number is negative or zero, return -1.

if any of the given numbers right most digit is of the number is lessthan 5, then round off it's value to the previous multiple of 10 otherwise if the right most digit of the number is greater or equal to 5, then round off to the next multiple of 10.

1. Define a method which accepts two value as arguments(an integer and boolean) and return the string indicating when the alarm should ring.

the first argument indicating day of the week encoded as 0=Sun, 1=Mon, 2=Tue, ...6=Sat, and a boolean indicating if we are on vacation or not.

Write the method with the following specifications

Name of method***ringAlarm()*** // which accepts two arguments, first indicating day of the week and second a boolean indiacting if we are on vacation.

Arguments: two arguments of type integer and boolean

Return Type: an string value

Specifications: The value returned by the method **ringAlarm()** is determined by the following rules

if the first argument value is not between 0 to 6, return "Invalid Inputs"

if the second value is not boolean value true or false, return "Invalid Inputs"

if the first argument value is between 1 to 5 indicating the week day's and second value is true indicating on vacation, return "10:00"

if the first argument value is between 1 to 5 indicating the week day's and second value is false indicating not on vacation, return "07:00"

if the first argument value is 0 or 6 indicating the weekend day's and second value is true indicating on vacation, return "OFF"

if the first argument value is 0 or 6 indicating the weekend day's and second value is false indicating not on vacation, return "10:00"

1. Define a method which accepts three boolean value as arguments and return true if any of the two values are true, other wise return false.

Write the method with the following specifications

Name of method ***countBoolean()*** // which accepts three boolean arguments, return true if any of the two values are true, else return false.

Arguments: Three arguments of type boolean

Return Type: A boolean value

Specifications: The value returned by the method countBoolean() is determined by the following rules

if b1 = true, b2 = true, b3 = true then, return true

if b1 = true, b2 = true, b3 = false then, return true

if b1 = true, b2 = false, b3 = false then, return false

if b1 = false, b2 = false, b3 = false then, return false

1. Define a method which returns a String containing natural numbers between a range of two numbers separated by a single space. Solve it using While loops.

Write the method with the following specifications

Name of method : ***getNaturalNumbers()***

Arguments : 2 Arguments of int type.

Return Type : A String value

Values must not be negative. If yes, then return -1 as string.

Values must not be 0. If yes, then return -2 as string.

Natural Numbers must be returned as one string with every value separated by single blankspace.

Consider that, the first argument value is less than the second argument number

1. Define a method which returns a String contraining  natural numbers between a range of two numbers separated by a single space.

Write the method with the following specifications

Name of method : ***getNumbersInRange()***

Arguments : 2 Arguments of int type

Return Type : A String value

Values must not be negative. If yes, then return -1 as string.

Values must not be same. If yes, then return -2 as string.

1st arguments value must not be greater than 2nd arguments value. If yes, then return -3 as string.

Exclude the first and last value and return the result as one string with every value separated by single blankspace.

1. Define a method which returns a String containing natural numbers between a given range separated by a single space.

Write the method with following specifications  
Name of method: **getNumbersInRange()** // accepts two integer value, return the number between the given range  
Arguments: Two arguments of integer type, start\_val and end\_val  
Return Type: A String value  
  
Example:  
Inputs: 10 1  
Output: 9 8 7 6 5 4 3 2  
  
Inputs: 3 0  
Output: 2 1   
  
Inputs: 3 2   
Output: "-4" // as the difference between 3 and 2 is 1

Specifications: The value returned by the method getNumbersInRange() is determined by the following rules:  
  
If the input values are negative, return "-1" as string  
If the input values are equal, return "-2" as string  
If the first argument value is less than second argument value, return "-3" as string  
If the difference between first argument and second argument is 1, return "-4" as string  
in other case return a string with every value separated by single blankspace.  
  
*Note: Numbers in range must not include start value and end value.*

1. Define a method which returns a string of natural numbers within a range of two numbers with a given step/increment factor.

Write the method with following specifications

Name of method : ***getNumbersInRange()***

Arguments : 3 arguments of type int   
// first argument as start value  
// second argument as end value  
// third argument is the step/increment value

Return Type : A String value

Values must not be negative. If yes, then return -1 as string.

Values must not be same. If yes, then return -2 as string.

1st value must not be greater than 2nd value. If yes, then return -3 as string.

Numbers in range must not include start and end value.

Numbers in range must be returned as one string with every value separated by single blankspace.

For Example:  
If input values are 10 30 2  
Output: 11 13 15 17 19 21 23 25 27 29

1. Define a method which returns a string of all numbers between 1 and the given input value.

Write the method with following specifications

Name of method : ***getFourPerLine()***

Arguments : 1 argument of type int

Return Type : A String value

Value must not be negative. If yes, then return -1 as string.

Value must not be 0. If yes, then return -2 as string.

Value must not be greater than 99. If yes, then return -3 as string.

Numbers in range must be returned as one string with every value separated by single blankspace.

Ensure a new line after every set of 4 values.

For Example:  
In Input: 12  
Output:  
1 2 3 4   
5 6 7 8  
9 10 11 12

1. Define a method which accepts 2 numeric arguments and returns a box of the same size as a string with asterisk '\*' symbol.

Write the method with following specifications

Name of method : ***createBoxPattern()***

Arguments : 2 Integer Arguments

// 1st argument represents Number of Rows.

// 2nd argument represents Numbe of Columns.

Return Type : A String value

Value must not be negative. If yes, then return -1 as string.

2) Value must not be 0. If yes, then return -2 as string.

1st value is rows and 2nd value is columns.

Box must be created using star symbol separated using a single blankspace for e.g.  1st value=4 & 2nd value=5 then output must be

**\*  \*  \*  \*  \***

**\*              \***

**\*              \***

**\*  \*  \*  \*  \***

1. Define a method which accepts 1 numeric argument and returns a String of stars (\*).

For instance if the given input is 3,then First line must have One star, Second Line Two stars, Third line Three Stars.

Write the method with following specifications

Name of method :***createStarPattern()***

Arguments : 1 Integer Argument           // Represents Number of Rows.

Return Type : A String value

Value must not be negative. If yes, then return -1 as string.

Value must not be 0. If yes, then return -2 as string.

Pattern must be created using star symbol separated by single blankspace.

**Example :**

Input : 4

**Output:**

**\***

**\*  \***

**\*  \*  \***

**\*  \*  \*  \***

1. Define a method which accepts 1 numeric argument and returns a pattern of numbers as a string.

Write the method with following specifications

Name of method : ***NumberPattern4()***

Arguments : 1 Integer Argument // Represents Number of Rows.

Return Type : A String value

Value must not be negative. If yes, then return -1 as string.

Value must not be 0. If yes, then return -2 as string.

Value is rows.

Pattern must be created using numbers separated by single blankspace

**Example :**

**Input:** 5

**Output:**

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

1. Define a Method that checks for whether a given input is a prime number or not, and return a string.

Write the method with following specifications

Name of method : ***checkPrime()***

Arguments : 1 Integer Argument

Return Type : A String value

Value must not be negative. If yes, then return -1 as string.

Value must not be 0 or 1. If yes, then return -2 as string.

If value is a prime number, then return true as string otherwise return false as string.

1. Define a Method that checks for whether a given input is a palindrome number or not, and return a string.

Write the method with following specifications

Name of method : ***checkPalindrome()***

Arguments : 1 Integer Argument

Return Type : A String value

Value must not be negative. If yes, then return   -1 as string.

Value must not be from 0 to 9. If yes, then return -2 as string.

If value is a palindrome, then return true as string otherwise return false as string.

1. Define a method which accepts a 4-digit value as number and checks whether the number is Armstrong.

Write the method with the following specifications:   
Name of method **checkArmStrong()** // which accepts an integer value as argument and return the String as specified below.  
Arguments: one argument of type integer  
Return Type: an String value

Specifications: The value returned by the method checkArmStrong() is determined by the following rules:  
  
If the input value is negative, return "-1" as string.  
If the input value is not an 4-digit number, return "-2" as string.  
If the input value is an 4-digit number and is an Armstrong, return "Armstrong Number" as string otherwise return "Not Armstrong Number" as string.  
  
*Note: The string values you are returning are case sensitive.*

1. Define a method which accepts a value as number and returns the factorial of the value.

Write the method with the following specifications

Name of method***getFactorial()*** which accepts an integer value as argument and return the factorial of the given value.

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method **getFactorial()** is determined by the following rules

Value must not be negative. If yes, then return   -1

Value must not be 0. If yes, then return -2

1. Define a method which accepts a integer value as argument and return the factors of the given value.

Write the method with the following specifications:  
  
Name of method **getFactors()** // which accepts an integer value as argument and return a String.  
Arguments: one argument of type integer  
Return Type: String value

Input: 28  
Output: "1 2 4 7 14 28"  
  
Input: -20  
Output: "-1"  
  
Input: 0  
Output: "-2"  
  
Specifications: The value returned by the method getFactors() is determined by the following rules:  
  
1) If the value is negative, return "-1" as string  
2) If the value is zero, return "-2" as string  
3) In other case, return the string, concatenating all the factors of the given number separating each factor with a blank space.

1. Define a method which accepts a value as number and returns the sum of factors of the value.

Write the method with the following specifications

Name of method ***getSumOfFactors()*** // which accepts an integer value as argument and return an Integer.

Arguments: one argument of type integer

Return Type: integer value

Specifications: The value returned by the method ***getSumOfFactors()*** is determined by the following rules

Value must not be negative. If yes, then return   -1

Value must not be 0. If yes, then return -2.

1. Define a method which accepts 2 values as number and returns the even numbers between the 2 values.

*Note: If the first argument value is less than second value, generate even numbers from first value to second value. If the second value is less than first value, generate even numbers from second value to first.*

Write the method with the following specifications

Name of method ***getEvenNumbers()*** // which accepts 2 integer values as arguments and return the even numbers between the range.

Arguments: Two arguments of type integer

Return Type: String value

Specifications: The value returned by the method ***getEvenNumbers()*** is determined by the following rules

Values must not be negative. If yes, then return   -1 as string.

Value must not be 0 or same. If yes, then return -2 as string.

The values must be returned as a single string where each value is separated by a single blank space.

For Example:  
Input: 10 20  
Output: 10 12 14 16 18 20  
Input: 20 10  
Output: 10 12 14 16 18 20

1. Define a method which returns the square of the given value

Write the method with the following specifications

Name of method***getSquare()***

Arguments: one argument of type integer

Return Type: an integer value

Specifications: The value returned by the method ***getSquare()*** is determined by the following rules

if the given number is 0, return -1

if the given number is negative value, return -2

for any positive value return square of the number

1. Define a method which accepts 3 values as number and checks and returns the name of the type of triangle generated.

Write the method with the following specifications

Name of method***findTriangle()***

Arguments: three arguments of type integers

Return Type: String value

Specifications: The value returned by the method ***findTriangle()*** is determined by the following rules

Values must not be 0. If yes, then return -1 as string.

Values must not be negative. If yes, then return       -2 as string.

Sum of two sides must be greater than the third side. If no, then return -3 as string

If its a triangle with valid sides, then return as string  whether the triangle formed is EQUILATERAL, ISOSCELES or SCALENE.

1. Define a method which accepts 2 values as number and returns the prime numbers between the values.

Write the method with the following specifications

Name of method ***getPrimeNumbers()***

Arguments: two arguments of type integers

Return Type: String value

Specifications: The value returned by the method ***getPrimeNumbers()*** is determined by the following rules

Value must not be negative. If yes, then return -1 as string.

1st value must not be greater than or equal to 2nd value. If yes, then return -2 as string.

Prime Numbers must be returned as one string with every value separated by single blank space.

1. Define a method which accepts two integer values as arguments and return the sum of prime numbers between the given range.

*Note: 1 is not a prime*

Write the method with the following specifications

Name of method ***getPrimeNumbersSum()*** // which accepts two integer values as argument and return the sum of all prime number between the range.

Arguments: two arguments of type integer

Return Type: an integer value

Specifications: The value returned by the method ***getPrimeNumbersSum()*** is determined by the following rules

In any of the input value is negavie, return -1

In any of the input value is zero, return -2

If the first value is greater than or equal to second value, return - 3.

1. Define a method which accepts 2 values as strings and returns a weaved string.

Write the method with the following specifications

Name of method ***getWeavedString()***

Arguments: two arguments of type strings

Return Type: an string value

Specifications: The value returned by the method getWeavedString() is determined by the following rules

Values must not be blank. If yes, then return -1 as string.

If value1 is greater than value2 in length, then return a concatenated string which looks like value2+value1+value2.

If value1 is smaller than value2 in length, then return a concatenated string which looks like value1+value2+value1.

If value1 equal to value2 in length, then return a concatenated string which contains each character from both the values at the same position. For e.g."Hello"  "Hello"  result = "HHeelloo"

1. Define a method which accepts an array of numbers and return the sum of unique numbers from the array.

Write the method with the following specifications

Name of method ***sumOfArray()***

Arguments: one argument of type integer array

Return Type: an integer value

Specifications: The value returned by the method ***sumOfArray()*** is determined by the following rules

Array must not be null or length must not be 0. If yes, then return -1

Array must not contain negative values. If yes, then return -2

Input: {1, 2, 3, 2, 4, 1}

Output: 10

1. Define a method that adds two numbers.

Write the method with following specifications

Name of method : ***getSum()***

Arguments : Two integers

Return Type : String

This method must take two integers and return sum of those integers.

If input is negative or zero, return "Error".

If input is positive, return the sum of integers

1. A class **FillMultiples** is given to you.

Implement logic for the following method

method: public static int[] getMultiplesArray(int number).

This method should create a new integer array of size 10 and fill it with the multiples (multiples 1 to 10) of the given number.

and return the array.

If the given number is negative or zero, return null.

1. Define the method which accepts two integer arguments, an integer array and integer value, which return the number of time the given value exist in the array.

Note: The elements in an array can be negative

Name of Method: getCount() // Which returns the number of time the given value exist in the array.  
Arugments: Two arguments of an integer array and integer value to search  
Return Type: an integer value  
  
Specifications: The value returned by the method sumOfArray() is determined by the following rules  
  
If the array is empty or null, return -1  
In other case return the integer value, indicating the number of times the element exist in the array

1. A class **RussianMultiplication** is given to you.

Implement the following method in that class

public static int getProduct(int num1, int num2)

If number is not positive then return -1.

Return the product of the two numbers.

Calculate the product using Russian multiplication process.

**Russian Multiplication**

The Russian multiplication, also called the Russian peasant algorithm, uses a halving and doubling method to multiply whole numbers

When halving, discard any remainder. Just put the quotient in the halving column.

When the number in the halving column is 1, cross out all rows that have an even number in the halving column

The answer is found by adding the remaining numbers in the doubling column

**Example # 1:** 11 × 12

Halving                              Doubling

   11                    ×                  12

   5                      ×                  24

   2                      ×                  48 ---> Discard this since 2 is even

   1                      ×                  96

The product of 11 and 12 is: 12 + 24 + 96 = 132