19. Sum of Digits in a String

Write code to get the sum of all the digits present in the given string.

Include a class **UserMainCode** with a static method **sumOfDigits** which accepts string input.

Return the sum as output. If there is no digit in the given string return -1 as output.

Create a class **Main** which would get the input and call the

static method **sumOfDigits** present in the UserMainCode.

Input and Output Format:

Input consists of a string.

Output is a single integer which is the sum of digits in a given

string. Refer sample output for formatting specifications.

Sample Input 1:

good23bad4

Sample Output 1:

9

Sample Input 2:

good

Sample Output 2:

-1

Main:

```
=====
Import java.util.*;
public class Main {
public static void main (String [] args)
{

}
```

UserMainCode:

```
============
import java.util.*;
import java.text.*;

public class UserMainCode{
public static void sumOfDigits(String s1) {

}}
```

20. String Concatenation

Write code to get two strings as input and If strings are of same length simply append them together and return the final string. If given strings are of different length, remove starting characters from the longer string so that both strings are of same length then append them together and return the final string.

Include a class **UserMainCode** with a static method **concatstring** which accepts two string input.

The return type of the output is a string which is the concatenated string. Create a class **Main** which would get the input and call the static

method **concatstring** present in the UserMainCode.

Input and Output Format:

Input consists of two

strings. Output is a string.

Refer sample output for formatting specifications.

Sample Input 1:

Hello

hi

Sample Output 1:

lohi

Sample Input 2:

Hello

Delhi

Sample Output 2:

HelloDelhi

```
import java.util.*;
public class Main {
public static void main (String [] args)
{

}
```

UserMainCode:

```
============
import java.util.*;
import java.text.*;

public class UserMainCode{
public static void concatstring(String s1,String s2){
```

}}

21. Color Code

Write a program to read a string and validate whether the given string is a valid color code based on the following rules:

- Must start with "#" symbol
- Must contain six characters after #
- It may contain alphabets from A-F or digits from 0-9

Include a class **UserMainCode** with a static method **validateColorCode** which accepts a string. The return type (integer) should return 1 if the color is as per the rules else return -1. Create a Class Main which would be used to accept a String and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a string.

Output consists of a string (Valid or Invalid).

Refer sample output for formatting specifications. **Sample Input 1:**

#FF9922

Sample Output 1:

Valid

Sample Input 2:

#FF9(22

Sample Output 2:

Invalid

Main

```
====
import java.util.*;
public class Main {
public static void main (String [] args)
{

}
```

UserMainCode:

```
============
import java.util.*;
import java.text.*;

public class UserMainCode{
public static int validateColorCode(String s1) {
}}
```

22. Three Digits

Write a program to read a string and check if the given string is in the format "CTS

-XXX" where XXX is a three digit number.

Include a class **UserMainCode** with a static method **validatestrings** which accepts a string.

The return type (integer) should return 1 if the string format is correct else

return -1. Create a Class Main which would be used to accept a String and call

the static method present in UserMainCode.

Input and Output Format:

Input consists of a string.

Output consists of a string (Valid or Invalid).

Refer sample output for formatting

specifications. **Sample Input 1:**

CTS-215

Sample Output 1:

Valid

Sample Input 2:

CTS-2L5

Sample Output 2:

Invalid

Main

```
====
import java.util.*;
public class Main {
public static void main (String [] args)
{

}
```

UserMainCode:

```
============
import java.util.*;
import java.text.*;

public class UserMainCode{
public static int validatestrings(String s1){

}}
```

23. Removing Keys from HashMap

Given a method with a HashMap<Integer,string> as input. Write code to remove all the entries having keys multiple of 4 and return the size of the final hashmap.

Include a class **UserMainCode** with a static method **sizeOfResultandHashMap** which accepts hashmap as input.

The return type of the output is an integer which is the size of the resultant hashmap.

Create a class **Main** which would get the input and call the static method **sizeOfResultandHashMap** present in the UserMainCode. **Input and Output Format:**

First input corresponds to the size of the hashmap. Input consists of a hashmap<integer,string>.

Output is an integer which is the size of the hashmap. Refer sample output for formatting specifications.

SampleInput1:
3
2
hi
4
Hell

12

Helloworld

SampleOutput1:
1

SampleInput2:
3

2
hi
4
sdfsdf
3
asdf

SampleOutput2:
2

Main

```
====
import java.util.*;
public class Main {
public static void main (String [] args)
{

}
```

UserMainCode:

```
============
import java.util.*;
import java.text.*;

public class UserMainCode{
public static int sizeOfResultandHashMap(HashMap<Integer,String> hm){
}}
```