46. Odd Digit Sum

Write a program to input a String array. The input may contain digits and alphabets ("de5g4G7R"). Extract odd digits from each string and find the sum and print the output. For example, if the string is "AKj375A" then take 3+7+5=15 and not as 375 as digit.
Include a class **UserMainCode** with a static method **oddDigitSum** which accepts a string array and the size of the array. The return type (Integer) should return the sum.
Create a Class Main which would be used to accept Input Strings and call the static method present in UserMainCode.
Assume maximum length of array is 20.
Input and Output Format:
Input consists of an integer n, corresponds to the number of strings, followed by n Strings.
Output consists of an Integer.
Refer sample output for formatting specifications.
Sample Input :
3
cog2nizant1
al33k
d2t4H3r5
Sample Output :
15
(1+3+3+3+5)

Main
======

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

}
```

UserMainCode
============

```
import java.util.*;
public class UserMainCode {
public static int oddDigitSum(String[] s1){

}}
```

47. Unique Number

Write a program that accepts an Integer as input and finds whether the number is Unique or not. Print Unique if the number is "Unique", else print "Not Unique".
Note: A Unique number is a positive integer (without leading zeros) with no duplicate digits.For example 7, 135, 214 are all unique numbers whereas 33, 3121, 300 are not.
Include a class **UserMainCode** with a static method **getUnique** which accepts an integer. The return type (Integer) should return 1 if the number is unique else return -1.
Create a Class Main which would be used to accept Input Integer and call the static method present in UserMainCode.
Input and Output Format:
Input consists of an integer .
Output consists of a String ("Unique" or "Not Unique"). Refer sample output for formatting specifications.
Sample Input 1:
123
Sample Output 1:
Unique
Sample Input 2:
33
Sample Output 2:
Not Unique

Main

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

}
```

UserMainCode

```
============
import java.util.*;
public class UserMainCode {
public static int getUnique(int n){

}}
```

48. Sum of Lowest marks

Given input as HashMap, value consists of marks and rollno as key.Find the sum of the lowest three subject marks from the HashMap.
Include a class **UserMainCode** with a static method **getLowest** which accepts a Hashmap

with marks and rollno.
The return type of the output is the sum of lowest three subject marks.
Create a class **Main** which would get the input and call the static method **getLowest** present
in the UserMainCode.
Input and Output Format:
First line of the input corresponds to the HashMap size.
Input consists a HashMap with marks and rollno.
Output is an integer which is the sum of lowest three subject marks. Refer sample output for formatting specifications.
Sample Input 1:
5
1
54
2
85
3
74
4
59
5
57
Sample Output 1:
170

Sample Input 2:
4
10
56
20
58
30
87
40
54
Sample Output 2:
168

Main

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

}
```

UserMainCode

```
============
import java.util.*;
public class UserMainCode {
public static int getLowest(HashMap<Integer, Integer> h1){

}}
```

50. Repeating set of characters in a string

Get a string and a positive integer n as input .The last n characters should repeat the number of times given as second input.Write code to repeat the set of character from the given string.
Include a class **UserMainCode** with a static method **getString** which accepts a string and an
integer n as input.
The return type of the output is a string with repeated n characters.
Create a class **Main** which would get the input and call the static method **getString** present
in the UserMainCode.

Input and Output Format:

Input consists a string and a positive integer
n. Output is a string with repeated
characters.
Refer sample output for formatting specifications.

Sample Input 1:

Cognizant
3

Sample Output 1:

Cognizantantantant

Sample Input 2:

myacademy
2

Sample Output 2:

Myacademymymy

Main

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

}
```

UserMainCode

```
============
import java.util.*;
public class UserMainCode {
public static String getString(String input, int n){
}}
```

51. Finding the day of birth

Given an input as date of birth of person, write a program to calculate on which day (MONDAY,TUESDAY....) he was born store and print the day in Upper Case letters.
Include a class **UserMainCode** with a static method **calculateBornDay** which accepts a string
as input.
The return type of the output is a string which should be the day in which the person was born.
Create a class **Main** which would get the input and call the
static method **calculateBornDay** present in the
UserMainCode.
Input and Output Format:
NOTE: date format should be(dd-MM-yyyy)
Input consists a date string.
Output is a string which the day in which the person was
born. Refer sample output for formatting specifications.
Sample Input 1:
29-07-2013
Sample Output 1:
MONDAY
Sample Input 2:
14-12-1992
Sample Output 2:
MONDAY

Main

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

}
```

UserMainCode

```
============
import java.util.*;
public class UserMainCode {
public static String calculateBornDay(String input){

}}
```