5.1.dateofbirth

Write a program to validate the Date of Birth given as input in String format (MM/dd/yyyy) as per the validation rules given below. Return true for valid dates else return false.
1. Value should not be null
2. month should be between 1-12, date should be between 1-31 and year should be a four digit number.
Include a class UserMainCode with a static method ValidateDOB which accepts the string. The return type is TRUE / FALSE.
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 TRUE / FALSE.
Refer sample output for formatting specifications.
Sample Input 1:
12/23/1985
Sample Output 1:
TRUE
Sample Input 2:
31/12/1985
Sample Output 2:
FALSE
UserMainCode

```
import java.text.SimpleDateFormat;
import java.util.Date;
public class UserMainCode {
public static Boolean ValidateDOB(String str){
 Boolean b=false;
 SimpleDateFormat sdf=new SimpleDateFormat("MM/dd/yyyy");
 sdf.setLenient(false);
 try
 {
 Date d1=sdf.parse(str);
 return b=true;
 }
 catch(Exception e)
 {
 return b=false;
 }
}
}
```

Main

```
package DateOfBirth;
import java.util.Scanner;
public class Main {
public static void main(String[] args)
{
String str=new String();
Scanner sc=new Scanner(System.in);
str=sc.nextLine();
Boolean b=UserMainCode.ValidateDOB(str);
if(b==true)
 System.out.println("TRUE");
if(b==false)
 System.out.println("FALSE");
```

```
}
}
```

```
31/12/1985
```

FALSE

3.digits2

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

Sample Input 2: 698 Sample Output 2: 5

UserMainCode

```
public class UserMainCode
{
public static int getDigitSum(int n)
 {
 int sum = 0 ;
 while(n>10)
 {
 int a = 0 ; sum = 0;
 while(n!=0)
 {
 a = n%10;
 sum+=a;
 n=n/10;
 }
 n=sum;
 }
 return sum;
 }
}
```

Main

```
import java.util.Scanner;
public class Main {
public static void main(String[] args)
 {
 Scanner s=new Scanner(System.in);
 int a=s.nextInt();
 int sum=UserMainCode.getDigitSum(a);
 System.out.println(sum);
 }
}698
5
```

output

6.2.grade calculator (code:28)

A School wants to assign grades to its students based on their marks. You have been assigned as the programmer to
automate this process. You would like to showcase your skills by creating a quick prototype. The prototype consists of the
following steps:
1. Read student details from the User. The details would include roll no, mark in the given order. The datatype for id
is integer, mark is integer.
2. You decide to build a hashmap. The hashmap contains roll no as key and mark as value.
3. BUSINESS RULE:
1. If Mark is greater than or equal to 80 store medal as ""GOLD"".
2. If Mark is less then to 80 and greater than or equal to 60 store medal as ""SILVER"".
3 .If Mark is less then to 60 and greater than or equal to 45 store medal as ""BRONZE"" else store ""FAIL"".
Store the result in TreeMap in which Roll No as Key and grade as value.
4. You decide to write a function calculateGrade which takes the above hashmaps as input and returns the treemap as
output. Include this function in class UserMainCode.
Create a Class Main which would be used to read employee details in step 1 and build the two hashmaps. Call the static
method present in UserMainCode.
Input and Output Format:
Input consists of employee details. The first number indicates the size of the students. The next two values indicate the roll
id, mark.
Output consists of a single string.
Refer sample output for formatting specifications.
Sample Input 1:

```
2
1010
80
100
40
```

Sample Output 1:

```
100
FAIL
1010
GOLD
```

UserMainCode

```
import java.util.Iterator;
import java.util.HashMap;
import java.util.TreeMap;
public class UserMainCode
{
public static TreeMap<Integer,String>calculateGrade(HashMap<Integer,Integer>hm)
{
TreeMap<Integer,String>tm=new TreeMap<Integer,String>();
Iterator<Integer> it=hm.keySet().iterator();
while(it.hasNext())
{
int id=it.next();
int mark=hm.get(id);
if(mark>=80)
tm.put(id,"GOLD");
else if(mark<80 && mark>=60)
tm.put(id,"SILVER");
else if(mark<60 && mark>=45)
tm.put(id,"BRONZE");
else
tm.put(id,"FAIL");
}
return tm;
}}Main
import java.util.HashMap;
import java.util.Iterator;
import java.util.TreeMap;
import java.util.Scanner;
public class Main {
public static void main(String []args){
Scanner sc=new Scanner(System.in);
int s=sc.nextInt();
HashMap<Integer,Integer>hm=new HashMap<Integer,Integer>();
for(int i=0;i<s;i++)
{
hm.put(sc.nextInt(),sc.nextInt());
}
TreeMap<Integer,String>tm=new TreeMap<Integer,String>();
tm=UserMainCode.calculateGrade(hm);
Iterator<Integer> it=tm.keySet().iterator();
for(int i=0;i<s;i++)
{
int n=it.next();
String fac=tm.get(n);
System.out.println(n);
System.out.println(fac);
}
}
}
1010
80
100
40
100
FAIL
1010
GOLD
```

6.1.grade calculator (code:53)

A School wants to give assign grades to its students based on their marks. You have been assigned as the programmer to automate this process. You would like to showcase your skills by creating a quick prototype. The prototype consists of the following steps:
Read student details from the User. The details would include name, mark in the given order. The datatype for name is string, mark is float.
You decide to build a hashmap. The hashmap contains name as key and mark as value.
BUSINESS RULE:
1. If Mark is less than 60, then grade is FAIL.
2. If Mark is greater than or equal to 60, then grade is PASS.
Note: FAIL/PASS should be in uppercase.
Store the result in a new Hashmap with name as Key and grade as value.
4. You decide to write a function calculateGrade which takes the above hashmap as input and returns the hashmap as output. Include this function in class UserMainCode.
Create a Class Main which would be used to read student details in step 1 and build the hashmap. Call the static method present in UserMainCode.
Input and Output Format:
Input consists of student details. The first number indicates the size of the students. The next two values indicate the name, mark.
Output consists of a name and corresponding grade for each student.
Refer sample output for formatting specifications.
Sample Input 1:

```
3
Avi
76.36
Sunil
68.42
Raja
36.25
```

Sample Output 1:

```
Avi
PASS
Sunil
PASS
Raja
FAIL
```

UserMainCode

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.TreeMap;
public class UserMainCode {
 public static TreeMap<String, String>calculategrade(HashMap<String, Float>hm){
 TreeMap<String, String> tm = new TreeMap<String, String>();
 Iterator<String> it = hm.keySet().iterator();
 while(it.hasNext()){
 String name = it.next();
 float mark = hm.get(name);
 if(mark >= 60){
 tm.put(name,"PASS");
 } else if(mark <= 60){
 tm.put(name,"FAIL");
 }
 }
```

```
 return tm;
 }
}
```

Main

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Scanner;
import java.util.TreeMap;
public class Main {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 float s = scanner.nextFloat();
 scanner.nextLine();
 HashMap<String, Float> hm = new HashMap<String, Float>();
 for(int i = 0; i < s; i++){
 hm.put(scanner.nextLine(), scanner.nextFloat());
 scanner.nextLine();
 }
 TreeMap<String, String> tm = new TreeMap<String, String>();
 tm = UserMainCode.calculategrade(hm);
 Iterator<String> it = tm.keySet().iterator();
 for(int i = 0; i < s; i++){
 String n = it.next();
 String fac = tm.get(n);
 System.out.println(n);
 System.out.println(fac);
 }
 }
}
```

```
Raja
36.35
Avi
PASS
Raja
FAIL
Sunil
PASS
```

4.Max Substring

Write a program to accept two string inputs. The first being a source string and second one
a delimiter. The source string contains the delimiter at various locations. Your job is to return the substring with maximum number of characters. If two or more substrings have
maximim number of characters return the substring which appears first. The size of the delimiter is 1.
Include a class UserMainCode with a static method extractMax which accepts the string.
The return type (string) should be the max substring.
Create a Class Main which would be used to accept Input string and call the static method
present in UserMainCode.
Input and Output Format:
Input consists of a source string and delimiter.
Output consists of a string.
Refer sample output for formatting specifications.
Sample Input 1:
delhi-pune-patna
-
Sample Output 1:
Delhi\

UserMainCode

```
import java.util.StringTokenizer;
public class UserMainCode {
 public static String extractmax(String input1, String input2){
 int max = 0;
 String s3 = null;
 StringTokenizer st = new StringTokenizer(input1,input2);
 while (st.hasMoreTokens()){
 String s2 = st.nextToken();
 int n = s2.length();
 if(n > max){
 max = n;
 s3 = s2;
 }
 }
 return s3;
 }
}
```

Main

```
package Maxstring;
import java.util.Scanner;
public class Main {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 String input1 = scanner.nextLine();
 String input2 = scanner.nextLine();
 System.out.println(UserMainCode.extractmax(input1,input2));
 }
}
```

delhi-pune-patna, output : - delhi

2.Name Shrinking

Write a program that accepts a string as input and converts the first two names into dotseparated
initials and printa the output.
Input string format is 'fn mn ln'. Output string format is 'ln [mn's 1st character].[fn's 1st character]'
Include a class UserMainCode with a static method getFormatedString which accepts a string. The return type (String) should return the shrinked name.
Create a Class Main which would be used to accept Input String and call the static method present in UserMainCode.
Input and Output Format:
Input consists of a string.
Output consists of a String.
Refer sample output for formatting specifications.
Sample Input:
Sachin Ramesh Tendulkar
Sample Output:
Tendulkar R.S

UserMainCode

```
import java.util.StringTokenizer;
public class UserMainCode {
public static String getFormatedString(String s1) {
StringBuffer sb=new StringBuffer();
StringTokenizer st=new StringTokenizer(s1," ");
String s2=st.nextToken();
String s3=st.nextToken();
String s4=st.nextToken();
sb.append(s4).append(" ");
sb.append(s3.substring(0,1));
sb.append(".");
sb.append(s2.substring(0,1));
return sb.toString();
}
}
```

Main

```
import java.util.*;
public class Main {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
 String s1=sc.nextLine();
System.out.println(UserMainCode.getFormatedString(s1));
}
}
```

```
Sachin Ramesh Tendulkar
Tendulkar R.S
```

Set-1

1.String finder

Given three strings say Searchstring, Str1 and Str2 as input, write a program to find out if Str2 comes after Str1 in the Searchstring.
Include a class UserMainCode with a static method "stringFinder" that accepts 3 String arguments and returns an integer. The 3 arguments correspond to SearchString, Str1 and Str2. The function returns 1 if Str2 appears after Str1 in the Searchtring. Else it returns 2.
Create a class Main which would get 3 Strings as input and call the static method stringFinder present in the UserMainCode.
Input and Output Format:
Input consists of 3 strings.
The first input corresponds to the SearchString.
The second input corresponds to Str1.
The third input corresponds to Str2.
Output consists of a string that is either "yes" or "no"
Sample Input 1: Sample Output 1:
geniousRajKumarDev Yes
Raj
Dev
Sample Input 2: Sample Output 2:
geniousRajKumarDev No
Dev
Raj
UserMainCode

```
public class UserMainCode {
public static int stringFinder(String s1,String s2,String s3)
{
 String a1=s1.toLowerCase();
 String a2=s2.toLowerCase();
 String a3=s3.toLowerCase();
 if(a1.contains(a2)&&a1.contains(a3))
 {
 if(a1.indexOf(a2)<a1.indexOf(a3))
 {
 return 1;
 }
 else
 return 2;
 }
 return 0;
}}
```

Main

```
import java.util.*;
public class Main {
public static void main(String[] args) {
 Scanner s=new Scanner(System.in);
 String s1=s.next();
 String s2=s.next();
 String s3=s.next();
 int b=UserMainCode.stringFinder(s1, s2, s3);
if(b==1)
{
 System.out.println("Yes");
}
else
 System.out.println("No");
s.close();
 }
}
```

```
geniousRajKumarDev
Dev
Raj
No
```