

## Digits =II

### MAIN:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.println(UserMainCode.extractdigit(n));
        sc.close();
    }
}
```

### USERMAINCODE:

```
public class UserMainCode {
    public static int extractdigit(int n)
    {

        int n1 =n;
        int sum =0;
        while(n1>10)
        {
            int a=0;sum =0;

            while(n1>0)
            {
                a=n1%10;

                sum = sum + a;

                n1=n1/10;

            }
            n1=sum;
        }
        return sum;
    }
}
```

```
}
```

## STRING FINDER

### MAIN:

```
import java.util.Scanner;
```

```
public class Main {  
    public static void main(String[] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        String s1 = sc.next();  
        String s2 = sc.next();  
        String s3 = sc.next();  
        int n = UserMainCode.stringfinder(s1,s2,s3);  
        if(n == 1)  
        {  
            System.out.println("yes");  
        }  
        else  
        {  
            System.out.println("no");  
        }  
        sc.close();  
    }  
}
```

```
}
```

### USERMAINCODE:

```
public class UserMainCode {  
    public static int stringfinder(String s1,String s2,String s3)  
    {  
        String a=s1.toLowerCase();  
        String b=s2.toLowerCase();  
        String c=s3.toLowerCase();  
        if(a.contains(b) && a.contains(c))  
        {  
            if(a.indexOf(b)< a.indexOf(c))  
            {  
                return 1;  
            }  
        }  
    }  
}
```

```

        else
        {
            return 0;
        }
    }
    return 0;
}
}

```

## NAME SHRINKING

### MAIN:

```

import java.util.Scanner;
public class Main {

    public static void main(String[] args)
    {
        String k = new String();
        Scanner sc = new Scanner(System.in);
        k= sc.nextLine();
        System.out.println(UserMainCode.extractstring(k));
        sc.close();
    }
}

```

### USERMAINCODE:

```

import java.util.StringTokenizer;

public class UserMainCode {

    public static String extractstring(String s)
    {
        StringBuffer q = new StringBuffer();
        StringTokenizer st = new StringTokenizer(s," ");
        String a = st.nextToken();
        String b = st.nextToken();
        String c = st.nextToken();
    }
}

```

```

        q.append(c).append(" ");
        q.append(b.substring(0,1));
        q.append(".");
        q.append(a.substring(0,1));
        return q.toString();
    }
}

```

## MAX SUBSTRING

### MAIN

```

import java.util.Scanner;

public class Main {
    public static void main (String [] args)
    {
        Scanner sc = new Scanner(System.in);
        String ss = sc.nextLine();
        String s = sc.nextLine();
        System.out.println(UserMainCode.extractmax(ss,s));
        sc.close();
    }
}

```

### USERMAINCODE

```

import java.util.StringTokenizer;

public class UserMainCode {
    public static String extractmax(String ss,String s)
    {
        String r =null;
        int max =0;
        StringTokenizer st = new StringTokenizer(ss,s);
        while(st.hasMoreTokens())
        {
            String n = st.nextToken();
            int e = n.length();
            if(e > max)
            {

```

```

        r=n;
        max=e;
    }

}
return r;

}

}

```

## GRADE CALCULATOR 1

### MAIN

```

import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Scanner;

public class Main {
    public static void main (String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int i=0;
        LinkedHashMap<String,Float> hm = new LinkedHashMap<String,Float>();
        for(i=0;i<n;i++)
        {
            String s = sc.next();
            float f = sc.nextFloat();
            hm.put(s, f);
        }
        UserMainCode r = new UserMainCode();
        LinkedHashMap<String,String> hm2 = new LinkedHashMap<String,String>();
        hm2 = UserMainCode.calculategrade(hm);
        Iterator<String> it = hm2.keySet().iterator();
        while(it.hasNext())
        {
            String ss = it.next();
            String dd = hm2.get(ss);
            System.out.println(ss);
            System.out.println(dd);
        }
    }
}

```

```
    }
}
```

### **USERMAINCODE**

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Scanner;

public class UserMainCode {
    public static LinkedHashMap<String, String>
    calculategrade(LinkedHashMap<String,Float> hm)
    {
        LinkedHashMap<String,String> hm2 = new LinkedHashMap<String,String>();
        String r = new String();
        Iterator<String> it = hm.keySet().iterator();
        while(it.hasNext())
        {
            String s = it.next();
            float a = hm.get(s);
            if( a >= 60)
            {
                r = "PASS";
            }
            else if(a <60)
            {
                r= "FAIL";
            }
            hm2.put(s,r);
        }

        return hm2;
    }
}
```

## **GRADE CALCULATOR 2**

### **MAIN**

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.HashMap;
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);
} } }
```

### **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;
    }
}
```

## DATE VALIDATION

### MAIN

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s= sc.next();
        int b = UserMainCode.getvalues(s);
        if(b==1)
            System.out.println("Valid");
        else
            System.out.println("Invalid");
    }
}
```

### USERMAINCODE



```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
public class UserMainCode {
public static int getvalues(String s) {
if(s.matches("[0-9]{2}[.]{1}[0-9]{2}[.]{1}[0-9]{4}"))
{
SimpleDateFormat sdf=new SimpleDateFormat("dd.MM.yyyy");
sdf.setLenient(false);
try
{
Date d1=sdf.parse(s);
return 1;
} catch (ParseException e) {
return -1;
}}
else if(s.matches("[0-9]{2}/[1]{1}[0-9]{2}/[0-9]{4}"))
{
SimpleDateFormat sdf=new SimpleDateFormat("dd/MM/yyyy");
sdf.setLenient(false);
try
{
Date d1=sdf.parse(s);
return 1;
} catch (ParseException e) {
return -1;
}}
else if(s.matches("[0-9]{2}-[1]{1}[0-9]{2}-[0-9]{4}"))
{
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
sdf.setLenient(false);
try
{
Date d1=sdf.parse(s);
return 1;
} catch (ParseException e) {
return -1;
}}
else
return -1;
}
}
```

