1) Find out if the given number is an *Armstrong number*.

Logic: - If **153** is the Supplied value, then $1^3 + 5^3 + 3^3 = 1+125+27 = 153$
This is the same as supplied value hence it is an Armstrong number.

Armstrong.java

```
1  package Assignment.java;
2  import java.util.Scanner;
3  public class Armstrong {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7          int n,r,sum=0;
8          Scanner sc = new Scanner(System.in);
9          n=sc.nextInt();
10         sc.close();
11         int temp=n;
12         while(n>0) {
13             r=n%10;
14             n=n/10;
15             sum=sum+r*r*r;
16         }
17         if(temp==sum) {
18             System.out.println("Armstrong number");
19         }
20         else {
21             System.out.println("Not an Armstrong number");
22         }
23     }
24
25 }
26
```

Problems Javadoc Declaration Console

```
<terminated> Armstrong [Java Application] C:\Users\psritula\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe  (07-
153
Armstrong number
```

2) Find out all the ***Armstrong numbers*** *falling* in the range of **100-999**

Armstrong.java | ArmstrongnumberRange.java

```java
package Assignment.java;

public class ArmstrongnumberRange {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

                // TODO Auto-generated method stub

                for(int k=100;k<=999;k++) {
                    int n=k,r=0,sum=0;
                    while(n>0) {
                        r=n%10;
                        sum=sum+(r*r*r);
                        n=n/10;

                    }

                if(sum==k) {
                    System.out.println(k+" is armstrong number");
                }
                }
            }

        }
```

Problems | Javadoc | Declaration | Console

<terminated> ArmstrongnumberRange [Java Application] C:\Users\psritula\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (07-Dec-2022, 8:38:11 pm – 8:38:12 pm) [pid

```
153 is armstrong number
370 is armstrong number
371 is armstrong number
407 is armstrong number
```

3) Find out the simple as well as the compound interest of supplied value

Armstrong.java | ArmstrongnumberRange.java | CompoundInterest.java

```java
package Assignment.java;
import java.util.Scanner;
public class CompoundInterest {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the principal : ");
        double principal = input.nextDouble();
        System.out.println("Enter the rate: ");
        double rate=input.nextDouble();
        System.out.println("Enter the time: ");
        double time=input.nextDouble();
        System.out.println("Enter number of times interest is compounded: ");
        int number = input.nextInt();
        double interest=principal * (Math.pow((1+rate/100),(time * number)))-principal;
        System.out.println("Principal: "+principal);
        System.out.println("Interest Rate: "+rate);
        System.out.println("Time Duration: "+time);
        System.out.println("Number of Time interest compound: "+number);
        System.out.println("compound Interesst: "+interest);

        // TODO Auto-generated method stub

    }

}
```

```
Enter the principal :
1000
Enter the rate:
10
Enter the time:
3
Enter number of times interest is compounded:
1
Principal: 1000.0
Interest Rate: 10.0
Time Duration: 3.0
Number of Time interest compound: 1
compound Interesst: 331.00000000000045
```

4) Supply marks of three subject and declare the result, result declaration is based on below conditions:

Condition 1: -All subjects marks is greater than 60 is Passed

Condition 2: -Any two subjects marks are greater than 60 is Promoted

Condition 3: -Any one subject mark is greater than 60 or all subjects' marks less than 60 is failed.

Armstrong.java | ArmstrongnumberRange.java | CompoundInterest.java | Subjects.java

```java
package Assignment.java;
import java.util.Scanner;
public class Subjects {
    public String declareResults(double subj1,double subj2,double subj3) {
        double sum = subj1+subj2+subj3;
        if((sum<60 || (subj1>60 && subj2<60 && subj3 <60))||(sum<60 || (subj2>60 && subj1<60 && subj3<60))||(sum<60 || (subj3>6
        return "failed";
        }
        else if(sum>60 && ((subj1+subj2<=60) && (subj2+subj3<=60)&& subj1+subj3<=60))
            return "Passed";
        else
            return "passed\npromoted";
    }
    public static void main(String[] args) {
        double subj1,subj2,subj3;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the marks of subject1: ");
        subj1=sc.nextDouble();
        System.out.println("Enter the marks of subject2: ");
        subj2=sc.nextDouble();
        System.out.println("Enter the marks of subject3: ");
        subj3=sc.nextDouble();
        Subjects resultDeclaration = new Subjects();
        System.out.println(resultDeclaration.declareResults(subj1,subj2,subj3));
        }

}

```

```
Enter the marks of subject1:
100
Enter the marks of subject2:
99
Enter the marks of subject3:
67
passed
promoted
```

5) Calculate the income tax on the basis of following table.

Note:-Assume slab is consider for Male, Female as well as Senior citizen

Slab	Income Range	Tax payable in Percentage
Slab A	0-1,80,000	Nil
Slab B	1,81,001-3,00,000	10%
Slab C	3,00,001-5,00,000	20%
Slab D	5,00,001-10,00,000	30%

Accept CTC from user and display tax amount

```java
package Assignment.java;
import java.util.Scanner;
public class TaxAmount {
    double tax=0;
    public double calculateTaxAmount(int ctc) {
        if(ctc>0 && ctc<=180000) {
            tax=0;
        }
        else if(ctc >= 180001 && ctc <= 300000) {
            tax =(ctc*10)/100;
        }
        else if(ctc>=3000001 && ctc<=500000) {
            tax=(ctc*20)/100;
        }
        else if(ctc>=500001 && ctc <= 1000000)
            tax=(ctc*30)/100;
        return tax;
    }
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            int ctc;
            System.out.println("Enter your CTC: ");
            ctc=sc.nextInt();
            TaxAmount taxAmount = new TaxAmount();
            double tax;
            tax=taxAmount.calculateTaxAmount(ctc);
            System.out.println("Tax payable : "+tax);

    }}
```

```
Enter your CTC:
180023
Tax payable : 18002.0
```

6) Consider a CUI based application, where you are asking a user to enter his Login name and password, after entering the valid user-id and password it will print the message "Welcome" along with user name. As per the validation is concerned, the program should keep a track of login attempts. After three attempts a message should be flashed saying "Contact Admin" and the program should terminate.

```java
package Assignment.java;
import java.util.Scanner;
public class Login {
    String userId = "Ajay",password="password";
    int loginAttempt=3;
    public String loginUser(String user, String pass) {
        if(user.equals(userId)&& pass.equals(password)) {
            return "yes";
        }
        else {
            return "no";
        }}
    public static void main(String[] args) {
        Login login = new Login();
        String userId,password;
        Scanner sc = new Scanner(System.in);
        int loginAttempt =0;
        while(true) {
            System.out.println("Enter userId");
            userId = sc.next();
            System.out.println("Enter password");
            password=sc.next();
            String res = login.loginUser(userId, password);
            if(res.equals("yes")){
                System.out.println("You have entered wrong credential 3 times");
                System.out.println("Contact Admin");
                break;}
            System.out.println("You have entered wrong credentials ,Please enter the right credentials.");
}}}
```

```
Enter userId
Ajay
Enter password
password
You have entered wrong credential 3 times
Contact Admin
```

7) There is an Array which is of the size 15, which may or may not be sorted. You should write a program to accept a number and search if it in contained in the array

.

Example:

5	12	14	6	78	19	1	23	26	35	37	7	52	86	47

Value to be search is 19

```java
package Assignment.java;



public class SearchArray {
      public boolean searchArray(int[] arr,int toCheckValue){
          boolean valueFound=false;
          for(int i=0;i<arr.length;i++){
              if(arr[i]==toCheckValue)
                  valueFound=true;
          }
          return valueFound;
      }



  public static void main(String[] args) {
       int arr[] = {5,12,14,6,78,19,1,23,26,35,37,7,52,86,47};
       int valueToCheck = 19;
       SearchArray searchArray = new SearchArray();
       if (searchArray.searchArray(arr,  valueToCheck)) {
           System.out.println("element is not present in the array");
       }
    }
}

```

```
element is not present in the array
```

8) Using the above table write method apply sorting using **Bubble Sort**.

```java
package Assignment.java;
class Bubblesort {
    public int[] bubbleSort(int arr[]) {

        for(int i=0;i<arr.length-1;i++){
            for(int j=0;j<arr.length-1-i;j++){
                if(arr[j]>arr[j+1]){

                    int temp = arr[j];
                    arr[j]= arr[j+1];
                    arr[j+1]=temp;
                }
            }
        }
        return arr;
    }
    public static void main(String args[]) {
        int arr[] = {5,12,14,6,78,19,1,23,26,35,37,7,52,86,47};
        Bubblesort bubbleSort = new Bubblesort();
        int sortedArray[] =bubbleSort.bubbleSort(arr);
        for(int i:sortedArray){
            System.out.print(i+" ");
        }
}
}
```

```
<terminated> Bubblesort [Java Application] C:\Users\psritula\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe  (07
1 5 6 7 12 14 19 23 26 35 37 47 52 78 86
```

9) Accept the marks of three students for the subject say A, B, C. Find the total scored and the average in all the subjects. Also Find the Total and Average scored by students in each respective Subject.

```java
package Assignment.java;
public class Student {
    int subjectA,subjectB,subjectC;
    public Student(){}
    public Student(int a,int b,int c){
        this.subjectA = a;
        this.subjectB = b;
        this.subjectC = c;
            }
          public int [] subjectWiseMarks(Student[] student,String sub){
          int [] arr = new int[3];
                if(sub.equals("A")){
                    for(int i=0;i<student.length;i++)
                        arr[i]=student[i].subjectA;
                }
                else if(sub.equals("B")){
                    for(int i=0;i<student.length;i++)
                        arr[i]=student[i].subjectB;
                }
                else{
                    for(int i=0;i<student.length;i++)
                        arr[i]=student[i].subjectC;
                }
                return arr;
            }
            public int subjectTotalByStudents(int[] marks){
                int sum=0;
                for(int i=0;i<marks.length;i++){
                    sum+=marks[i];
                }
                 sum+=marks[i];
                }
                return sum;
            }
            public double subjectAverageByStudents(int [] marks) {
                double sum=0.0;
                for(int i=0;i<marks.length;i++) {
                    sum+=marks[i];
                }
                return sum/3;
            }
            public int studentsTotalMarksInAllSubjects(Student[] students){
                int total=0;
                for(int i=0;i<students.length;i++){
                    total = total+students[i].subjectA+students[i].subjectB+students[i].subjectC;
                }
                return total;
            }
        public double studentsAverageMarksInAllSubjects(Student[] students) {
            int total=0;
            for(int i=0;i<students.length;i++) {
                total=total+students[i].subjectA+students[i].subjectB+students[i].subjectC;
            }
            return total/3;
            }
        public int subjectATotalByStudents(int[] marks) {
            int total =0;
            for(int i=0;i<marks.length;i++) {
                total+=marks[i];
            }
```

```
58          }
59          return total;
60      }
61      public int subjectBTotalByStudents(int[] marks) {
62          int total=0;
63          for(int i=0;i<marks.length;i++) {
64              total+=marks[i];
65          }
66          return total;
67          }
68      public int subjectCTotalByStudents(int[] marks) {
69          int total=0;
70          for(int i=0;i<marks.length;i++) {
71              total+=marks[i];
72          }
73          return total;
74          }
75      public double subjectAAverageByStudents(int[] marks) {
76          int total=0;
77          for(int i=0;i<marks.length;i++) {
78              total+=marks[i];
79          }
80          return total/3;
81      }
82      public double subjectBAverageByStudents(int[] marks) {
83          int total=0;
84          for(int i=0;i<marks.length;i++) {
85              total+=marks[i];
```

```
85              total+=marks[i];
86          }
87          return total/3;
88      }
89      public double subjectCAverageByStudents(int[] marks) {
90          int total=0;
91          for(int i=0;i<marks.length;i++) {
92              total+=marks[i];
93          }
94          return total/3;
95      }
96      public int getSubjectA() {
97          return subjectA;
98      }
99      public void setSubjectA(int subjectA) {
100         this.subjectA=subjectA;
101     }
102     public int getSubjectB() {
103         return subjectB;
104     }
105     public void setSubjectB(int subjectB) {
106         this.subjectB=subjectB;
107     }
108     public int getSubjectC() {
109         return subjectC;
110     }
111     public void setSubjectC(int subjectC) {
112         this.subjectC=subjectC;
113     }
```

```
    }

public static void main(String[] args) {
    Student s1 = new Student(10,20,30);
    Student s2 = new Student(10,20,30);
    Student s3 = new Student(10,20,30);
    Student [] students = {s1,s2,s3};
    Student student = new Student();
    int [] marksA = new int[3];
    for(int i=0;i<students.length;i++) {
        marksA[i]=10;
    }
    int [] marksB = new int[3];
    for(int i=0;i<students.length;i++) {
        marksA[i]=20;
    }
    int [] marksC = new int[3];
    for(int i=0;i<students.length;i++) {
        marksA[i]=30;
    }
     System.out.println(student.studentsTotalMarksInAllSubjects(students));
        System.out.println(student.studentsAverageMarksInAllSubjects(students));
        System.out.println(student.subjectTotalByStudents(marksA));
        System.out.println(student.subjectAverageByStudents(marksA));
        System.out.println(student.subjectBTotalByStudents(marksB));
        System.out.println(student.subjectBAverageByStudents(marksB));
        System.out.println(student.subjectCTotalByStudents(marksC));
    }
}
```

Writable | Smart Insert | 81 : 10 : 2362

```
<terminated> Student [Java Application] C:\Users\psritula\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe  (08-De
180
60.0
90
30.0
0
0.0
0
```