Write a program to input two numbers and find their product.

Source Code:

```
import java.util.Scanner;
class FindingProduct
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter any two numbers to multiply:");
      int a=sc.nextInt();
      int b=sc.nextInt();
      int c=a*b;
      System.out.println("The result is:"+c);
   }
}
```

Output:

```
Enter any two numbers to multiply:
2 3
The result is:6
```


Write a program to input a number and print its multiplication table.

Source Code:

```
import java.util.Scanner;
class MultiplicationTable
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter a number to print its multiplication table:");
      int a=sc.nextInt();
      for(int i=1;i<=10;i++)
      {
         System.out.println(a+ "X" +i + "="+(a*i));
      }
   }
}
```

Output:

```
Enter a number to print its multiplication table:8
8X1=8
8X2=16
8X3=24
8X4=32
8X5=40
8X6=48
8X7=56
8X8=64
8X9=72
8X10=80
```


Write a program to enter the price of apple per dozen and find the price of 10 apples.

Source Code:

```
import java.util.Scanner;
class FindingPrice
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the price of apple per dozen:");
      float p=sc.nextFloat();
      float a=p/12;
      float tp=a*10;
      System.out.println("Price of 10 apples:"+tp);
   }
}
```

Output:

```
Enter the price of apple per dozen:
120
Price of 10 apples:100.0
```


Write a program that takes character as input and displays whether it is vowel or consonant.

Source Code:

```
import java.util.Scanner;
class VowelConsonant
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a character to check vowel/consonant:");
      char a=sc.next().charAt(0);
      if(a>='a'&& a<='z' || a>='A' && a<='Z')
      {
         if(a=='a'||a=='e'||a=='i'||a=='o'||a=='u'||a=='A'||a=='E'||a=='I'||a=='O'||a=='U')
         System.out.println("Given Character is Vowel");
         else
         System.out.println("Given Character is Consonant");
      }
      else
      System.out.println("Given input is neither vowel nor consonant");
   }
}
```

Output:

```
Enter a character to check vowel/consonant:
I
Given Character is Vowel
```

Write a program to find area of triangle.

Source Code:

```
import java.util.Scanner;
class Area
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter the base and height:");
      float b=sc.nextFloat();
      float h=sc.nextFloat();
      float area=(b*h)/2;
      System.out.println("Area :"+area);
   }
}
```

Output:

```
Enter the base and height:
2 5
Area :5.0
```


Write a program that checks whether the character is in uppercase or lowercase.

Source Code:

```
import java.util.Scanner;
class UpperLower
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter a character to check Uppercase or Lowercase:");
      char a=sc.next().charAt(0);
      if(a>='a'&&a<='z'||a>='A'&&a<='Z')
      {
         if(a>='a'&&a<='z')
         System.out.println("Lowercase");
         else
         System.out.println("Uppercase");
      }
      else
      System.out.println("Invalid Input - Not a Character");
   }
}
```

Output:

```
Enter a character to check Uppercase or Lowercase:I
Uppercase
```

Write a program to demonstrate the use of all escape sequences available in Java.

Source Code:

```
class EscapeSequencesDemo {
    public static void main(String[] args) {
        // Demonstrating escape sequences
        System.out.println("Demonstrating all escape sequences in Java:");

        // Newline
        System.out.print("Hi\nHello");

        // Carriage return
        System.out.println("Hello, World!\rJava");

        // Tab
        System.out.println("Column1\tColumn2\tColumn3");

        // Backspace
        System.out.println("Lion\b\bar");

        // Single quote
        System.out.println("It\'s a beautiful day!");

        // Double quote
        System.out.println("She said, \"Hello!\"");

        // Backslash
        System.out.println("SDC\\BIM\\2080");
    }
}
```

Write a program to input a three-digit number and reverse its digits.

Source Code:

```
import java.util.Scanner;
class Reverse
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter a 3 digit number:");
      int a=sc.nextInt();
      int first,second,last;
      if(a>=100 && a<=999)
      {
         first=a/100;
         second=a/10;
         second=second%10;
         last=a%10;
         System.out.println("Reverse is:"+last+second+first);
      }
      else
      System.out.println("Invalid input (Not a 3 digit number)");
   }
}
```

Output:

```
Enter a 3 digit number:123
Reverse is:321
```


Write a program to print the greatest among three numbers using if else statement.

Source Code:

```
import java.util.Scanner;
class Greatest
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter any 3 numbers:");
      int a=sc.nextInt();
      int b=sc.nextInt();
      int c=sc.nextInt();
      if(a>b && a>c)
      System.out.println(a+"is greatest");
      else if(b>a && b>c)
      System.out.println(b+"is greatest");
      else
      System.out.println(c+"is greatest");
   }
}
```

Output:

```
Enter any 3 numbers:8 9 2
9is greatest
```


Write a program to print the greatest among three numbers without using if statement (Use ternary operator).

Source Code:

```
import java.util.Scanner;
class Greatest
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter any 3 numbers: ");
      int a=sc.nextInt();
      int b=sc.nextInt();
      int c=sc.nextInt();
      System.out.println((a>b && a>c)? (a + " is greatest") :
                  (b>a && b>c)? (b + " is greatest") :
                  (c + " is greatest"));
   }
}
```

Output:

```
Enter any 3 numbers: 8 2 1
8 is greatest
```


Write a program to input marks in 5 subjects and compute results as:

Percentage	Result
80-100	Distinction
60-80	First Division
50-60	Second Division
40-50	Third Division
Below 40	Fail

Note: Passing in each subject is mandatory to pass the overall exam.

Source Code:

```
import java.util.Scanner;
class Result
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter marks in 5 subjects:");
      int m1=sc.nextInt();
      int m2=sc.nextInt();
      int m3=sc.nextInt();
      int m4=sc.nextInt();
      int m5=sc.nextInt();
      float percentage;
      percentage=(m1+m2+m3+m4+m5)/5;
      System.out.println("Percentage:"+percentage);
      if(m1>=40 && m2>=40 && m3>=40 && m4>=40 && m5>=40)
      {
         if(percentage>=80)
         System.out.println("Result: Distinction");
         else if(percentage>=60)
         System.out.println("Result: First Division");
         else if(percentage>=50)
         System.out.println("Result: Second Division");
```

```
            else
            System.out.println("Result: Third Division");
        }
        else
        System.out.println("Result: Fail");
    }
}
```

Output:

```
Enter marks in 5 subjects:75 80 99 68 72
Percentage:78.0
Result: First Division
```


Write a program to create a menu driven program as follows:

1. Find Area of Square.
2. Find Area of Rectangle
3. Close the program.
Enter your choice:
Note: *The program should terminate only when user chooses "Close the program".*

Source Code:

```
import java.util.Scanner;
class MenuDriven
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      while(true)
      {
         System.out.println("1. Find Area of Square \n 2. Find Area of Rectangle \n 3. Close the program");
         System.out.print("Enter your choice:");
         int choice=sc.nextInt();
         switch(choice)
         {
            case 1:
            {
               int l;
               System.out.print("Enter length of square:");
               l=sc.nextInt();
               System.out.println("Area is :"+(l*l));
               System.out.print("\n");
               break;
            }
            case 2:
            {
```

```
            int l,b;
            System.out.print("Enter length and breadth of rectangle:");
            l=sc.nextInt();
            b=sc.nextInt();
            System.out.println("Area is :"+(l*b));
            System.out.print("\n");
            break;
        }
        case 3:
        {
            System.out.println("Program Terminated");
            return;
        }
        default: System.out.println("Invalid choice");
      }
    }
  }
}
```

Output:

```
1. Find Area of Square
 2. Find Area of Rectangle
 3. Close the program
Enter your choice:1
Enter length of square:4
Area is :16


1. Find Area of Square
2. Find Area of Rectangle
 3. Close the program
Enter your choice:2
Enter length and breadth of rectangle:3 4
```

```
Area is :12


1. Find Area of Square
 2. Find Area of Rectangle
 3. Close the program
Enter your choice:3
Program Terminated
```


Write a program that shows whether the switch case statement in java is fall through or not.

Source Code:

```
import java.util.Scanner;
class SwitchCase
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter from the options to print as follows: \n 1. Hello \n 2. Hi \n 3. Bye \n 4. Good job");
      System.out.print("Enter your choice:");
      int choice=sc.nextInt();
      switch(choice)
      {
         case 1:
         System.out.println("Hello");
         break;
         // It is not fall through as there is break
         case 2:
         System.out.println("Hi");
         break;
         // It is not fall through as there is break
         case 3:
         System.out.println("Bye");
         // It is fall through as there is no break
         case 4:
         System.out.println("Good job");
         // It is not fall through as there is no break
         default:
         System.out.println("Invalid Choice");
         /*If 3 is choosen as choice then all the cases from 3 onwards and default statement is executed */
```

```
        }
    }
}
```

Output:

```
Enter from the options to print as follows:
 1. Hello
 2. Hi
 3. Bye
 4. Good job
Enter your choice:3
Bye
Good job
Invalid Choice
```


Write a program that checks whether the entered number is palindrome or not using while loop.

Source Code:

```
import java.util.Scanner;
class PalindromeCheck
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter a number to check whether it is palindrome or not:");
      int n=sc.nextInt();
      int org=n,rev=0;
      while(n!=0)
      {
         int digit=n%10;
         rev=(rev*10)+digit;
         n=n/10;
      }
      if(org==rev)
      System.out.println("It is palindrome number");
      else
      System.out.println("It is not a palindrome number");
   }
}
```

Output:

```
Enter a number to check whether it is palindrome or not:1001
It is palindrome number
```

Write a program to check whether the entered number is prime or not.

Source Code:

```
import java.util.Scanner;
class PrimeCheck
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter a number to check whether it is prime number or not:");
      int n=sc.nextInt();
      int a=2,b=0;
      while(a<n)
      {
         if(n%a==0)
         b++;
         a++;
      }
      if(b>0)
      System.out.println("It is not a prime number");
      else
      System.out.println("It is a prime number");
   }
}
```

Output:

```
Enter a number to check whether it is prime number or not:17
It is a prime number
```


Write a program to print the sum of natural numbers up to n.

Source Code:

```
import java.util.Scanner;
class SumofNaturalNumbers
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the value of n:");
      int n=sc.nextInt();
      int a=1,sum=0;
      while(a<=n)
      {
         sum=sum+a;
         a++;
      }
      System.out.println("Sum is:"+sum);
   }
}
```

Output:

```
Enter the value of n:10
Sum is:55
```


Write a program to print the following pattern using while loop:

1
12
113
1114
11115

Source Code:

```
class Pattern
{
   public static void main(String[] args)
   {
      int i=1;
      while(i<=5)
      {
         int j=1;
         while(j<=i)
         {
            if(j==i)
            System.out.print(i);
            else
            System.out.print(1);
            j++;
         }
         System.out.print("\n");
         i++;
      }
   }
}
```


Write a program to print the following pattern using loop:

1
12
123
1234
12345

Source Code:

```
class Pattern
{
   public static void main(String[] args)
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<=i;j++)
         {
            System.out.print(j);
         }
         System.out.print("\n");
      }
   }
}
```


Write a program to print the following pattern using loop:

5
54
543
5432
54321

Source Code:

```
class Pattern
{
   public static void main(String[] args)
   {
      int i,j;
      for(i=5;i>=1;i--)
      {
         for(j=5;j>=i;j--)
         {
            System.out.print(j);
         }
         System.out.print("\n");
      }
   }
}
```

Write a program to print the following pattern using loop:

```
    1
   12
  123
 1234
12345
```

Source Code:

```
class Pattern
{
   public static void main(String[] args)
   {
      int i,j,k,sp=4;
      for(i=1;i<=5;i++)
      {
         for(k=1;k<=sp;k++)
         {
            System.out.print(" ");
         }
         for(j=1;j<=i;j++)
         {
            System.out.print(j);
         }
         System.out.print("\n");
         sp--;
      }
   }
}
```


Write a program to print the following pattern using loop:

```
    1
   21
  321
 4321
54321
```

Source Code:

```
class Pattern
{
   public static void main(String[] args)
   {
      int i,j,k,sp=4;
      for(i=1;i<=5;i++)
      {
         for(k=1;k<=sp;k++)
         {
            System.out.print(" ");
         }
         for(j=i;j>=1;j--)
         {
            System.out.print(j);
         }
         System.out.print("\n");
         sp--;
      }
   }
}
```


Write a program to print the following pattern using loop:

```
    1
   121
  12321
 1234321
123454321
```

Source Code:

```
class Pattern
{
   public static void main(String[] args)
   {
      int a=1,sp=4;
      do
      {
         for(int k=1;k<=sp;k++)
         {
            System.out.print(" ");
         }
         System.out.print((a*a)+"\n");
         a=(a*10)+1;
         sp--;
      }
      while (a<=11111);
   }
}
```

Write a program to print the following pattern using loop:

```
* * * * *
*       *
*       *
*       *
* * * * *
```

Source Code:

```
class StarPattern
{
   public static void main(String[] args)
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<=5;j++)
         {
            if(i==1||i==5||j==1||j==5)
            System.out.print("*");
            else
            System.out.print(" ");
         }
         System.out.print("\n");
      }
   }
}
```


Write a program to print the following pattern using loop:

```
* * * * *
* *   * *
*   *   *
* *   * *
* * * * *
```

Source Code:

```
class StarPattern
{
   public static void main(String[] args)
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<=5;j++)
         {
            if((i==2 && j==3) || (i==3 && j==2) || (i==3 && j==4) || (i==4 && j==3))
            System.out.print(" ");
            else
            System.out.print("*");
         }
         System.out.print("\n");
      }
   }
}
```


Write a program to print the following pattern using for loop:

1
1 2
1 1 3
1 1 1 4
1 1 1 1 5

Source Code:

```
class Newpattern
{
   public static void main(String[] args)
   {
      int i,j;
      for(i=1;i<=5;i++)
      {
         for(j=1;j<i;j++)
         {
            System.out.print("1");
         }
         System.out.print(i);
         System.out.print("\n");
      }
   }
}
```


Write a program to create a menu driven program that performs the following task (coversions):

1. Feet to Inch.
2. Inch to Feet.
3. KG to Pounds
4. Pounds to KG
5. NPR to USD
6. USD to NPR
7. Exit
Enter your choice:
Note: *The program should terminate only when user chooses "Exit".*

Source Code:

```
import java.util.Scanner;
class MenuDrivenProgram
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      while(true)
      {
         System.out.println("1. Feet to Inch \n 2. Inch to Feet \n 3. KG to Pounds \n 4. Pounds to KG \n 5. NPR to USD \n 6. USD to NPR \n 7. Exit");
         System.out.print("Enter your choice:");
         int choice=sc.nextInt();
         switch(choice)
         {
           case 1:
            {
               float feet;
               System.out.print("Enter Feet:");
               feet=sc.nextFloat();
               System.out.println("Inch is :"+(feet*12));
               System.out.print("\n");
```

```
    break;
}
case 2:
{
    float inch;
    System.out.print("Enter Inch:");
    inch=sc.nextFloat();
    System.out.println("Feet is :"+(inch/12));
    System.out.print("\n");
    break;
}
case 3:
{
    float kg;
    System.out.print("Enter KG:");
    kg=sc.nextFloat();
    System.out.println("Pound is:"+(kg*2.20462));
    System.out.print("\n");
    break;
}
case 4:
{
    float pound;
    System.out.print("Enter Pound:");
    pound=sc.nextFloat();
    System.out.println("KG is:"+(pound/2.20462));
    System.out.print("\n");
    break;
}
case 5:
{
    float npr;
```

```
                System.out.print("Enter NPR:");
                npr=sc.nextFloat();
                System.out.println("USD is:"+(npr/133));
                System.out.print("\n");
                break;
            }
            case 6:
            {
                float usd;
                System.out.print("Enter USD:");
                usd=sc.nextFloat();
                System.out.println("NPR is:"+(usd*133));
                System.out.print("\n");
                break;
            }
            case 7:
            {
                System.out.println("Program Closed");
                return;
            }
            default: System.out.println("Invalid choice");
        }
    }
  }
}
```

Output:

```
1. Feet to Inch
2. Inch to Feet
3. KG to Pounds
4. Pounds to KG
5. NPR to USD
```

6. USD to NPR

7. Exit

Enter your choice:1

Enter Feet:1

Inch is :12.0

1. Feet to Inch

2. Inch to Feet

3. KG to Pounds

4. Pounds to KG

5. NPR to USD

6. USD to NPR

7. Exit

Enter your choice:2

Enter Inch:12

Feet is :1.0

1. Feet to Inch

2. Inch to Feet

3. KG to Pounds

4. Pounds to KG

5. NPR to USD

6. USD to NPR

7. Exit

Enter your choice:3

Enter KG:1

Pound is:2.20462

1. Feet to Inch

2. Inch to Feet

3. KG to Pounds

4. Pounds to KG

5. NPR to USD

6. USD to NPR

7. Exit

Enter your choice:4

Enter Pound:2

KG is:0.9071858188712795

1. Feet to Inch

2. Inch to Feet

3. KG to Pounds

4. Pounds to KG

5. NPR to USD

6. USD to NPR

7. Exit

Enter your choice:5

Enter NPR:266

USD is:2.0

1. Feet to Inch

2. Inch to Feet

3. KG to Pounds

4. Pounds to KG

5. NPR to USD

6. USD to NPR

7. Exit

Enter your choice:6

Enter USD:5

NPR is:665.0

1. Feet to Inch

2. Inch to Feet

3. KG to Pounds

```
4. Pounds to KG
5. NPR to USD
6. USD to NPR
7. Exit
Enter your choice:7
Program Closed
```

Write a program that converts binary to decimal and vice versa.

Source Code:

```
import java.util.Scanner;
class conversion
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      while(true)
      {
         System.out.println("1. Binary to Decimal \n 2. Decimal to Binary \n 3. Close the program");
         System.out.print("Enter your choice:");
         int choice=sc.nextInt();
         switch(choice)
         {
            case 1:
            {
               int n,r,decimal=0,base=1;
               System.out.print("Enter Binary Number:");
               n=sc.nextInt();
               while(n>0)
               {
                  r=n%10;
                  decimal=decimal+r*base;
                  n=n/10;
                  base=base*2;
               }
               System.out.println("Decimal Number is:"+decimal);
               System.out.print("\n");
               break;
            }
```

```
            case 2:
            {
               int n,r,b=0,base=1;
               System.out.print("Enter Decimal Number:");
               n=sc.nextInt();
               while(n>0)
               {
                  r=n%2;
                  b=b+r*base;
                  n=n/2;
                  base=base*10;
               }
               System.out.println("Binary Number is:"+b);
               System.out.print("\n");
               break;
            }
            case 3:
            {
               System.out.print("Program Closed");
               return;
            }
            default:System.out.print("Invalid Choice \n");
         }
      }
   }
}
```

Output:

```
1. Binary to Decimal
2. Decimal to Binary
3. Close the program
Enter your choice:1
```

```
Enter Binary Number:1011
Decimal Number is:11


1. Binary to Decimal
 2. Decimal to Binary
 3. Close the program
Enter your choice:2
Enter Decimal Number:11
Binary Number is:1011


1. Binary to Decimal
 2. Decimal to Binary
 3. Close the program
Enter your choice:3
Program Closed
```


Write a program to print Armstrong numbers from m to n, where m & n are input.

Source Code:

```
import java.util.Scanner;
class Armstrong
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the starting number (m):");
      int m=sc.nextInt();
      System.out.print("Enter the ending number (n):");
      int n=sc.nextInt();
      System.out.print("Armstrong numbers:");
      for(int i=m;i<=n;i++)
      {
         int sum=0;
         int number=i;
         // Calculating sum of cube of digits
         while(number>0)
         {
            int digit=number%10;
            sum+=Math.pow(digit,3);
            number=number/10;
         }
         // Checking condition of armstrong
         // Checking if original number is equal to sum of cube of its digit or not
         if(sum==i)
         System.out.print(i+" ");
      }
   }
```

```
}
```

Output:

```
Enter the starting number (m):1
Enter the ending number (n):1000
Armstrong numbers:1 153 370 371 407
```


Write a program to enter a number and evaluate as:

$$\frac{1}{1} - \frac{1}{2^2} + \frac{1}{3^3} - \frac{1}{4^4} + \cdots + \frac{1}{n^n}$$

Source Code:

```
import java.util.Scanner;
class Evaluate
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the value of n:");
      int n=sc.nextInt();
      float sum=0;
      for(int i=1;i<=n;i++)
      {
         if(i%2==1)
         sum+=(1/ Math.pow(i,i));
         else
         sum-=(1/ Math.pow(i,i));
      }
      System.out.print("Result : "+sum);
   }
}
```

Output:

```
Enter the value of n:7
Result : 0.7834305
```

Write a program to print maximum and minimum element from an integer array.

Source Code:

```
import java.util.Scanner;
class MaximumMinimum
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the number of elements in array:");
      int n=sc.nextInt();
      int a[]=new int[n];
      int i;
      System.out.print("Enter "+n+" integers in array:");
      for(i=0;i<a.length;i++)
      {
         a[i]=sc.nextInt();
      }
      int longest=a[0],shortest=a[0];
      for(i=0;i<a.length;i++)
      {
         if(a[i]>longest)
         longest=a[i];
         if(a[i]<shortest)
         shortest=a[i];
      }
      System.out.print("Longest:"+longest+"\nShortest:"+shortest);
   }
}
```


Output:

```
Enter the number of elements in array:9
Enter 9 integers in array:7 8 5 9 2 1 6 4 3
Longest:9
Shortest:1
```


Write a program to print largest & second largest number from integer array.

Source Code:

```
import java.util.Scanner;
class LargestSecondlargest
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the number of elements in the array: ");
      int n = sc.nextInt();
      int a[] = new int[n];
      System.out.print("Enter the elements of the array:");
      for(int i=0;i<n;i++)
      {
         a[i]=sc.nextInt();
      }
      // Sort the array in ascending order
      for(int i=0;i<a.length;i++)
      {
         for(int j=i+1;j<a.length;j++)
         {
            if(a[i]>a[j])
            {
               int temp=a[i];
               a[i]=a[j];
               a[j]=temp;
            }
         }
      }

      if (n>=2)
```

```
        {
            System.out.print("Largest Number: " + a[n - 1] + "\nSecond Largest Number: " + a[n - 2]);
        }


    }
}
```

Output:

```
Enter the number of elements in the array: 10
Enter the elements of the array:10 22 32 102 52 40 24 28 33 60
Largest Number: 102
Second Largest Number: 60
```


Write a program to print sum of elements of an array.

Source Code:

```
import java.util.Scanner;
class SumOfElements
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the number of elements in array:");
      int n=sc.nextInt();
      int a[]=new int[n];
      int i,sum=0;
      System.out.print("Enter "+n+" integers in array:");
      for(i=0;i<a.length;i++)
      {
         a[i]=sc.nextInt();
         sum=sum+a[i];
      }
      System.out.print("Sum of Elements in Array is:"+sum);
   }
}
```

Output:

```
Enter the number of elements in array:10
Enter 10 integers in array:1 2 3 4 5 6 7 8 9 10
Sum of Elements in Array is:55
```


Write a program to print only the palindrome numbers from an array.

Source Code:

```
import java.util.Scanner;
class PalindromeArrayCheck
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the number of elements in array:");
      int n=sc.nextInt();
      int a[]=new int[n];
      int i,sum=0;
      System.out.print("Enter "+n+" numbers in array:");
      for(i=0;i<a.length;i++)
      {
         a[i]=sc.nextInt();
      }
      System.out.print("Palindrome Numbers:");
      for(i=0;i<a.length;i++)
      {
         int org=a[i],rev=0;
         while(a[i]!=0)
         {
            int digit=a[i]%10;
            rev=(rev*10)+digit;
            a[i]=a[i]/10;
         }
         if(org==rev)
         System.out.print(org+" ");
      }
   }
```

```
}
```

Output:

```
Enter the number of elements in array:5
Enter 5 numbers in array:100 121 232 250 301
Palindrome Numbers:121 232
```


Write a program to take marks of 5 subjects in an array and generate percentage obtained.

Source Code:

```
import java.util.Scanner;
class Percentage
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[5];
      int i;
      float sum=0;
      System.out.print("Enter marks in 5 subjects:");
      for(i=0;i<5;i++)
      {
         a[i]=sc.nextInt();
         sum=sum+a[i];
      }
      System.out.print("Percentage Obtained:" +(sum/5));
   }
}
```

Output:

```
Enter marks in 5 subjects:85 90 92 95 75
Percentage Obtained:87.4
```


Write a program to print sum of all the elements of a matrix m × n

Source Code:

```
import java.util.Scanner;
class MatrixSum
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the row size(m):");
      int m=sc.nextInt();
      System.out.print("Enter the column size(n):");
      int n=sc.nextInt();
      int matrix[][]=new int[m][n];
      System.out.println("Enter the elements of matrix:");
      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            matrix[i][j]=sc.nextInt();
         }
      }
      int sum=0;
      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            sum=sum+matrix[i][j];
         }
      }
      System.out.print("Sum of all elements of matrix:"+sum);
   }
```

```
}
```

Output:

```
Enter the row size(m):2
Enter the column size(n):3
Enter the elements of matrix:
1 2 3
4 5 6
Sum of all elements of matrix:21
```


Write a program to print transpose of matrix.

Source Code:

```
import java.util.Scanner;
class Transpose
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the row size(m):");
      int m=sc.nextInt();
      System.out.print("Enter the column size(n):");
      int n=sc.nextInt();
      int matrix[][]=new int[m][n];
      System.out.println("Enter the elements of matrix:");
      int i,j;
      for(i=0;i<m;i++)
      {
         for(j=0;j<n;j++)
         {
            matrix[i][j]=sc.nextInt();
         }
      }
      System.out.println("Transpose Matrix:");
      for(i=0;i<m;i++)
      {
         for(j=0;j<n;j++)
         {
            System.out.print(matrix[j][i]+" ");
         }
         System.out.print("\n");
      }
```

```
    }
}
```

Output:

```
Enter the row size(m):3
Enter the column size(n):3
Enter the elements of matrix:
1 2 3
4 5 6
7 8 9
Transpose Matrix:
1 4 7
2 5 8
3 6 9
```


Write a program to find sum of two matrices

Source Code:

```
import java.util.Scanner;
class SumofMatrices
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the row size:");
      int m=sc.nextInt();
      System.out.print("Enter the column size:");
      int n=sc.nextInt();
      int matrix1[][]=new int[m][n];
      int matrix2[][]=new int[m][n];
      int summatrix[][]=new int[m][n];
      int i,j;
      System.out.println("Enter the elements of first matrix:");
      for(i=0;i<m;i++)
      {
         for(j=0;j<n;j++)
         {
            matrix1[i][j]=sc.nextInt();
         }
      }
      System.out.println("Enter the elements of second matrix:");
      for(i=0;i<m;i++)
      {
         for(j=0;j<n;j++)
         {
            matrix2[i][j]=sc.nextInt();
         }
```

```
        }
        System.out.println("Resultant Matrix:");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                summatrix[i][j]=matrix1[i][j]+matrix2[i][j];
                System.out.print(summatrix[i][j]+" ");
            }
            System.out.print("\n");
        }
    }
}
```

Output:

```
Enter the row size:2
Enter the column size:2
Enter the elements of first matrix:
1 2
3 4
Enter the elements of second matrix:
1 2
3 4
Resultant Matrix:
2 4
6 8
```


Write a program to print sum of diagonal elements of a matrix.

Source Code:

```
import java.util.Scanner;
class MatrixDiagonalSum
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the row size:");
      int m=sc.nextInt();
      System.out.print("Enter the column size:");
      int n=sc.nextInt();
      int matrix[][]=new int[m][n];
      System.out.println("Enter the elements of matrix:");
      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            matrix[i][j]=sc.nextInt();
         }
      }
      int sum=0;
      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            if(i==j)
            sum=sum+matrix[i][j];
         }
      }
      System.out.print("Sum of diagonal elements:"+sum);
```

```
    }
}
```

Output:

```
Enter the row size:3
Enter the column size:3
Enter the elements of matrix:
1 2 3
4 5 6
7 8 9
Sum of diagonal elements:15
```


Write a program to multiply two matrices.

Source Code:

```
import java.util.Scanner;
class MatrixMultiplication
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the row size for first matrix:");
      int row1=sc.nextInt();
      System.out.print("Enter the column size for first matrix:");
      int column1=sc.nextInt();
      int matrix1[][]=new int[row1][column1];
      System.out.print("Enter the row size for second matrix:");
      int row2=sc.nextInt();
      System.out.print("Enter the column size for second matrix:");
      int column2=sc.nextInt();
      int matrix2[][]=new int[row2][column2];
      int matrixmul[][]=new int[row1][column2];
      int i,j,k;
      if(column1==row2)
      {
         System.out.println("Enter the elements of first matrix:");
         for(i=0;i<row1;i++)
         {
            for(j=0;j<column1;j++)
            {
               matrix1[i][j]=sc.nextInt();
            }
         }
         System.out.println("Enter the elements of second matrix:");
```

```
        for(i=0;i<row2;i++)
        {
           for(j=0;j<column2;j++)
           {
              matrix2[i][j]=sc.nextInt();
           }
        }
        for(i=0;i<row1;i++)
        {
           for(j=0;j<column2;j++)
           {
              for(k=0;k<column1;k++)
              {
                 matrixmul[i][j]+=matrix1[i][k]*matrix2[k][j];
              }
           }
        }
        System.out.println("The Resultant Matrix is:");
        for(i=0;i<row1;i++)
        {
           for(j=0;j<column2;j++)
           {
              System.out.print(matrixmul[i][j]+" ");
           }
           System.out.print("\n");
        }
     }
     else
     System.out.print("Matrix Multiplication is not possible");
  }
}
```

Output:

```
Enter the row size for first matrix:2
Enter the column size for first matrix:2
Enter the row size for second matrix:2
Enter the column size for second matrix:2
Enter the elements of first matrix:
7 5
6 3
Enter the elements of second matrix:
2 1
5 1
The Resultant Matrix is:
39 12
27 9
```


Write a Java program to display all the even numbers from 1 to 500. [BIM – 2015 TU]

Source Code:

```
class DisplayEvenNumbers
{
   public static void main(String[] args)
   {
      System.out.println("Even Numbers from 1 to 500:");
      int i;
      for(i=1;i<=500;i++)
      {
         if(i%2==0)
         System.out.print(i+" ");
      }
   }
}
```

Output:

```
Even Numbers from 1 to 500:
2 4 6 8 10 12 14 16 18 ……. 496 498 500
```

Make an array of integers of size 30, store 30 integers, then display integers that are between 16 and 47. [BIM – 2016 TU]

Source Code:

```
import java.util.Scanner;
class ArrayofIntegers
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[30];
      int i;
      System.out.print("Enter 30 Integers:");
      for(i=0;i<30;i++)
      {
         a[i]=sc.nextInt();
         if(a[i]>16 && a[i]<47)
         System.out.print(a[i]+" ");
      }
   }
}
```

Output:

```
Enter 30 Integers:1 2 5 8 15 17 18 54 46 26 27 19 77 78 79 90 92 65 79 85 100 101 1000 240 500 550 705 876 657 784
17 18 46 26 27 19
```


Write a program that asks the user to enter numbers in an array of size 'n'. Then displays only the numbers that are divisible by 2 and 3. [BIM – 2017 TU]

Source Code:

```
import java.util.Scanner;
class ArrayofIntegers
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the value of n:");
      int n=sc.nextInt();
      int a[]=new int[n];
      int i;
      System.out.print("Enter "+n+" Numbers:");
      for(i=0;i<n;i++)
      {
         a[i]=sc.nextInt();
         if(a[i]%2==0 && a[i]%3==0)
         System.out.print(a[i]+" ");
      }
   }
}
```

Output:

```
Enter the value of n:7
Enter 7 Numbers:6 7 8 9 10 11 12
6 12
```


Write a program that asks the user to enter elements in a matrix of size m × n and then display the sum of elements. [BIM – 2018 TU]

Source Code:

```
import java.util.Scanner;
class SumOfElements
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the row size:");
      int m=sc.nextInt();
      System.out.print("Enter the column size:");
      int n=sc.nextInt();
      int matrix[][]=new int[m][n];
      System.out.println("Enter the elements of matrix:");
      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            matrix[i][j]=sc.nextInt();
         }
      }
      int sum=0;
      for(int i=0;i<m;i++)
      {
         for(int j=0;j<n;j++)
         {
            sum=sum+matrix[i][j];
         }
      }
      System.out.print("Sum of elements:"+sum);
```

```
    }
}
```

Output:

```
Enter the row size:2
Enter the column size:2
Enter the elements of matrix:
1 2
3 4
Sum of elements:10
```


Write a program that displays only the prime numbers between any two integers.

[BIM – 2021 TU]

Source Code:

```
import java.util.Scanner;
class PrimePrint
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the starting number:");
      int starting=sc.nextInt();
      System.out.print("Enter the ending number:");
      int ending=sc.nextInt();
      System.out.println("Prime Numbers:");
      for(int number=starting;number<ending;number++)
      {
         if(number>1)
         {
            int count=0;
            for(int i=2;i<=number/2;i++)
            {
               if(number%i==0)
               {
                  count++;
                  break;
               }
            }
            if(count==0)
               System.out.print(number+" ");
         }
      }
   }
```

```
}
```

Output:

```
Enter the starting number:1
Enter the ending number:10
Prime Numbers:
2 3 5 7
```


Write a program that takes 10 integers as input from user, stores them in an array and find the product of only the numbers which are less than 5.

[BIM – 2021 – Make Up - TU]

Source Code:

```
import java.util.Scanner;
class ArrayProduct
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10];
      int i,product=1;
      System.out.print("Enter 10 integers:");
      for(i=0;i<a.length;i++)
      {
         a[i]=sc.nextInt();
         if(a[i]<5)
         product=product*a[i];
      }
      System.out.print("Product is: "+product);
   }
}
```

Output:

```
Enter 10 integers:1 2 3 4 5 6 7 8 9 10
Product is: 24
```

Write a program which contains a class Rectangle with member variables length and breadth. Then create three instances(objects) of rectangle in main class and print area of rectangle whose length is greatest.

Source Code:

```
class Rectangle
{
   double length,breadth;
}
public class Area
{
   public static void main(String[] args)
   {
      Rectangle r1=new Rectangle();
      r1.length=3;
      r1.breadth=2;
      Rectangle r2=new Rectangle();
      r2.length=4;
      r2.breadth=3;
      Rectangle r3=new Rectangle();
      r3.length=5;
      r3.breadth=4;
      double area;
      if(r1.length>r2.length && r1.length>r3.length)
      area=r1.length*r1.breadth;
      else if(r2.length>r1.length && r2.length>r3.length)
      area=r2.length*r2.breadth;
      else
      area=r3.length*r3.breadth;
      System.out.print("Area of Rectangle:"+area);
   }
}
```

Output:

```
Area of Rectangle:20.0
```

Create a class TwoDpoint with member variable x, y (both double), create two instances of point and print distance between them. Create member method to initialize the point variable.

Source Code:

```
import java.util.Scanner;
class TwoDpoint
{
   double x,y;
   void setDimensions(double a,double b)
   {
      x=a;
      y=b;
   }
}
public class Ashesh
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      TwoDpoint p1=new TwoDpoint();
      TwoDpoint p2=new TwoDpoint();
      p1.setDimensions(1,3);
      p2.setDimensions(1,5);
      double m=Math.pow((p2.x-p1.x),2);
      double n=Math.pow((p2.y-p1.y),2);
      double s=m+n;
      System.out.print("Distance:"+Math.sqrt(s));
   }
}
```

Output:

```
Distance:2.0
```

Make class "Rectangle" with attributes length and breadth. The class contains methods computeArea and displayArea. Write a program with main method that creates two objects of Rectangle class and find their areas and display area of larger rectangle. [BIM – 2015 – TU]

Source Code:

```
import java.util.Scanner;
class Rectangle
{
   double length, breadth;
   void setDimensions(double length, double breadth)
   {
      this.length = length;
      this.breadth = breadth;
   }
   double computeArea()
   {
      return length*breadth;
   }
   void displayArea()
   {
      System.out.println("Area: " + computeArea());
   }
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      Rectangle rect1 = new Rectangle();
      Rectangle rect2 = new Rectangle();
      System.out.print("Enter length and breadth for first rectangle: ");
      rect1.setDimensions(sc.nextDouble(), sc.nextDouble());
      System.out.print("Enter length and breadth for second rectangle: ");
      rect2.setDimensions(sc.nextDouble(), sc.nextDouble());
      double area1 = rect1.computeArea();
```

```
        double area2 = rect2.computeArea();
        if (area1 > area2)
        System.out.println("First Rectangle has the larger area: " + area1);
        else if (area2 > area1)
        System.out.println("Second Rectangle has the larger area: " + area2);
        else
        System.out.println("Both rectangles have the same area: " + area1);
    }
}
```

Output:

```
Enter length and breadth for first rectangle: 5 4
Enter length and breadth for second rectangle: 5 5
Second Rectangle has the larger area: 25.0
```


Create a class Number with three integer instance variables x, y and z. The class will have one constructor to initialize instance variables. The class also will contain method getMax() method that will return the largest number. Create a class NumberDemo with main method that will create an object of Number and will print the largest number. [BIM – 2019 – TU]

Source Code:

```
class Number
{
   int x,y,z;
   Number(int a, int b, int c)
   {
      x=a;
      y=b;
      z=c;
   }
   int getMax()
   {
      if(x>y && y>z)
      return x;
      else if(y>x && x>z)
      return y;
      else
      return z;
   }
}
public class NumberDemo
{
   public static void main(String[] args)
   {
      Number E1=new Number(1,2,3);
      int m=E1.getMax();
      System.out.println("Maximum Number: "+m);
```

```
    }
}
```

Output:

```
Maximum Number: 3
```


Write a program to print numbers from 1 to n using recursion.

Source Code:

```
import java.util.Scanner;
public class Print
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the value of n:");
      int n=sc.nextInt();
      Print ob=new Print();
      ob.PrintNumbers(n,1);
   }
   void PrintNumbers(int n, int i)
   {
      if(i<=n)
      {
         System.out.print(i+" ");
         PrintNumbers(n,i+1);
      }
   }
}
```

Output:

```
Enter the value of n:5
1 2 3 4 5
```


Write a program to print numbers from n to 1 using recursion.

Source Code:

```
import java.util.Scanner;
public class Print
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the value of n:");
      int n=sc.nextInt();
      Print ob=new Print();
      ob.PrintNumbers(n,1);
   }
   void PrintNumbers(int n, int i)
   {
      if(i<=n)
      {
         System.out.print(n+" ");
         PrintNumbers(n-1,i);
      }
   }
}
```

Output:

```
Enter the value of n:5
5 4 3 2 1
```


Write a program to print sum of natural numbers upto n by using recursion.

Source Code:

```
import java.util.Scanner;
public class Print
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter the value of n:");
      int n=sc.nextInt();
      Print ob=new Print();
      System.out.print("The sum of natural number is: "+ob.sum(n));
   }
   public int sum(int n)
   {
      if(n==1)
      return 1;
      else
      return n+sum(n-1);
   }
}
```

Output:

```
Enter the value of n:10
The sum of natural number is: 55
```


Write a program to check whether the number is prime or not using recursion.

Source Code:

```
import java.util.Scanner;
public class PrimeCheck
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter any number: ");
      int n=sc.nextInt();
      PrimeCheck ob=new PrimeCheck();
      if (n<=1)
      {
         System.out.println(n + " is not a prime number");
      }
      else
      {
         int x=ob.prime(n,2);
         if(x==1)
         System.out.println(n + " is a prime number");
         else
         System.out.println(n + " is not a prime number");
      }
   }
   int prime(int y, int i)
   {
      if (i*i>y)
      {
         return 1;
      }
      else if (y%i==0)
```

```
        {
            return 0;
        }
        else
        {
            return prime(y,i+1);
        }
    }
}
```

Output:

```
Enter any number: 17
17 is a prime number
```

*Write a program to find the product of x*y using addition operator only.*

Source Code:

```
import java.util.Scanner;
class Multiplication
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter any two numbers to find it's product:");
      int x=sc.nextInt();
      int y=sc.nextInt();
      Multiplication ob=new Multiplication();
      System.out.print("Result is: "+ob.mul(x,y));
   }
   int mul(int x, int y)
   {
      if(x==1)
      return y;
      else if(y==1)
      return x;
      else
      return x+mul(x,y-1);
   }
}
```

Output:

```
Enter any two numbers to find it's product:2 3
Result is: 6
```

Write a program to print Fibonacci series upto n by using recursion.

Source Code:

```
import java.util.Scanner;
class Fibonacci
{
   public static int fibo(int n)
   {
      if(n<=1)
      return n;
      else
      return fibo(n-1)+fibo(n-2);
   }
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter a number:");
      int n=sc.nextInt();
      for(int i=0;i<n;i++)
      {
         System.out.print(fibo(i)+" ");
      }
   }
}
```

Output:

```
Enter a number:5
0 1 1 2 3
```


Write a program with overloaded method getarea which prints the area of square, rectangle and triangle.

Source Code:

```
class MethodOverloading
{
   static double getarea(double l)
   {
      return l*l;
   }
   static double getarea(double l, double b)
   {
      return l*b;
   }
   static double getareatriangle(double b,double h)
   {
      return 0.5*b*h;
   }
   public static void main(String[] args)
   {
      System.out.println("Area of Square:"+getarea(5));
      System.out.println("Area of Rectangle:"+getarea(5,4));
      System.out.println("Area of Triangle:"+getareatriangle(10,20));
   }
}
```

Output:

```
Area of Square:25.0
Area of Rectangle:20.0
Area of Triangle:100.0
```

Write a program that counts the number of objects created.

Source Code:

```
class Count
{
   static int a=0;
   Count()
   {
      a++;
   }
}
public class New
{
   public static void main(String[] args)
   {
      Count c1=new Count();
      Count c2=new Count();
      Count c3=new Count();
      System.out.print("The number of objects created is:"+Count.a);
   }
}
```

Output:

```
The number of objects created is:3
```

Create a class currency with member variables Rs. And Paisa (Both int). Create two object of currency and store their sum in another currency object. The addition of currency should be done in such a way that if paisa is greater than 99, Rs should be incremented by 1 & Paisa should be decremented by 100.

Source Code:

```
class Currency
{
   int Rs,Paisa;
   void CurrencyValue(int r,int p)
   {
      Rs=r;
      Paisa=p;
   }
   void CurrencySum(int r,int p)
   {
      Rs=r+(p/100);
      Paisa=p%100;
      System.out.print("Rupees:"+Rs+" "+"Paisa:" +Paisa);
   }
}
public class First
{
   public static void main(String[] args)
   {
      int r,p;
      Currency ob1=new Currency();
      ob1.CurrencyValue(125,100);
      Currency ob2=new Currency();
      ob2.CurrencyValue(35,80);
      Currency ob3=new Currency();
      r=ob1.Rs+ob2.Rs;
      p=ob1.Paisa+ob2.Paisa;
```

```
        ob3.CurrencySum(r,p);
    }
}
```

Output:

```
Rupees:161 Paisa:80
```

Create a base class Complex with member variables real and imaginary. Also include the member method called display to print the value of complex number. Then create a child class complex1 with member variables real & imaginary and member method which adds its real with parents real and its imaginary with parent's imaginary and returns the result as a complex type. Also include constructor that initializes its member variable values. Create an object of complex1 and call the add method on behalf of it to print the result.

Source Code:

```
class Complex
{
   int real,imaginary;
   Complex(int r,int i)
   {
      real=r;
      imaginary=i;
   }
   void display()
   {
      System.out.print("The value of Complex Number is: "+(real)+"+"+(imaginary)+"i");
   }
}
class Complex1 extends Complex
{
   Complex1(int r,int i,int r1,int i1)
   {
      super(r,i);
      real=r1;
      imaginary=i1;
   }
   public Complex add()
   {
      int totalreal=super.real+real;
```

```
        int totalimaginary=super.imaginary+imaginary;
        return new Complex(totalreal,totalimaginary);
    }
}
public class First
{
    public static void main(String[] args)
    {
        Complex1 ob=new Complex1(3,4,5,6);
        Complex x=ob.add();
        x.display();
    }
}
```

Output:

```
The value of Complex Number is: 10+12i
```


Create an interface called Calculate which has methods int add (int x, int y) and int diff (int x, int y) to perform addition and subtraction of numbers passed as arguments. Then define a class that implements interface calculate. [BIM – 2016 TU]

Source Code:

```
interface Calculate
{
   int add(int x, int y);
   int diff(int x, int y);
}
class Cal implements Calculate
{
   public int add(int x, int y)
   {
      return x+y;
   }
   public int diff(int x, int y)
   {
      return x-y;
   }
}
public class Calculation
{
   public static void main(String[] args)
   {
      Calculate ob=new Cal();
      System.out.println("The addition is:"+ob.add(5,6)+" & The subtraction is:"+ob.diff(6,4));
   }
}
```

Output:

```
The addition is:11 & The subtraction is:2
```

Create a class Student with instance variables roll no and two method to read and display the roll no. Then, create another class Test that inherits class Student, consisting of its own instance variables to hold the marks of two subjects and also methods to read and display the marks. Finally, create another class Result which inherits class Test. It also has its own instance variable total to hold off two marks scored by the student. Similarly, it has methods to calculate and display the total. Create some instances of above classes and demonstrate inheritance. [BIM – 2017 TU]

Source Code:

```
class Student
{
   int roll_no;
   public void ToRead(int r)
   {
      roll_no=r;
   }
   public void ToDisplayRollno()
   {
      System.out.println("The Roll Number of Student is:"+roll_no);
   }
}
class Test extends Student
{
   int marks1,marks2;
   public void ToRead(int m1,int m2)
   {
      marks1=m1;
      marks2=m2;
   }
   public void ToDisplayMarks()
   {
      System.out.println("The marks obtained by student:"+marks1+" "+marks2);
```

```
    }
}
class Result extends Test
{
    int total;
    public void ToCalculate(int t)
    {
        total=t;
    }
    public void ToDisplayTotal()
    {
        System.out.print("Total:"+total);
    }
}
public class HisabKitab
{
    public static void main(String[] args)
    {
        Result ob=new Result();
        ob.ToRead(9); // initializing roll number
        ob.ToDisplayRollno();
        ob.ToRead(80,70); // initializing marks
        ob.ToDisplayMarks();
        int t=ob.marks1+ob.marks2;
        ob.ToCalculate(t); // t = total marks
        ob.ToDisplayTotal();
    }
}
```

Output:

```
The Roll Number of Student is:9
The marks obtained by student:80 70
```

Total:150

Create an abstract class called Fmachine, having methods getdata() and putdata(). Derive a class Airplane, having instance variables code, name, capacity and methods getdata() and putdata() that overrides Fmachine's getdata() and putdata() to read and display the result. [BIM – 2018 TU]

Source Code:

```
import java.util.Scanner;
abstract class Fmachine
{
   abstract public void getdata();
   abstract public void putdata();
}
class Airplane extends Fmachine
{
   String name,code;
   int capacity;

   public void getdata()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter code, name and capacity:");
      code=sc.nextLine();
      name=sc.nextLine();
      capacity=sc.nextInt();
   }
   public void putdata()
   {
      System.out.println("Code:"+code+" Name:"+name+" Capacity:"+capacity);
   }
}
public class JavaSuru
{
   public static void main(String[] args)
```

```
    {
        Fmachine ob=new Airplane();
        ob.getdata();
        ob.putdata();
    }
}
```

Output:

```
Enter code, name and capacity:
ATR-901
Shree Airlines
10000
Code:ATR-901 Name:Shree Airlines Capacity:10000
```


Write a program to demonstrate example of nested try- catch. [BIM – 2015 TU]

Source Code:

```
class ExceptionDemo
{
   public static void main(String[] args)
   {
      try
      {
         int a=5/1;
         System.out.println(a);
         try
         {
            int b[]=new int[3];
            System.out.println(b[3]);
         }
         catch(ArithmeticException e)
         {
            System.out.println("Arithmetic Exception Occured");
         }
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
         System.out.println("Array Exception Occured");
      }
   }
}
```

Output:

```
5
Array Exception Occured
```

Write a program to catch the ArrayIndexOutOfBounds exception. [BIM – 2016 TU]

Source Code:

```
class ArrayExceptionDemo
{
   public static void main(String[] args)
   {
      try
      {
         int a[]={4,5,6,7,8};
         System.out.println(a[10]);
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
         System.out.println("Exception Occured");
      }
   }
}
```

Output:

```
Exception Occured
```


Write a program to demonstrate example of user defined exception. [BIM – 2017 TU]

Source Code:

```
import java.util.Scanner;
class UserDefinedExceptionDemo
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      try
      {
         System.out.print("Enter age:");
         int age=sc.nextInt();
         if(age<18)
         throw new VotingException();
         else
         System.out.print("You can vote");
      }
      catch(VotingException e)
      {
         System.out.println(e);
      }
   }
}
class VotingException extends Exception
{
   public String toString()
   {
      return("You cannot vote");
   }
}
```

Output:

```
Enter age:17

You cannot vote
```


Write a program to show the use of different access specifiers in Java.

Source Code:

```
class A
{
   protected int a=5;
   int b=4; // default more restriction than protected
   private int c=6; // highest restriction
   public int d=2; // lowest restriction
   public void result()
   {
      System.out.println(a+b+c+d); // can be accessed within the same class
   }
}
class B extends A
{
   public void Add()
   {
      System.out.println(a+b+d); // c cannot be accessed in sub class as it is private member
   }
}
public class Demo
{
   public static void main(String[] args)
   {
      B ob=new B();
      ob.Add();
   }
}
```

Output:

```
11
```

Write a program to create a class Mobile (type, Phone_no). Customize the exception such that if the user give phone_no having less than or greater than 10 digit, then the program has to throw an exception with the message "Invalid Phone number". [BIM – 2023 TU]

Source Code:

```
import java.util.Scanner;
class Mobile
{
   String type;
   long Phone_no;
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      try
      {
         System.out.print("Enter Phone Number:");
         long num=sc.nextLong();
         int count=0;
         while(num!=0)
         {
            num=num/10;
            count++;
         }
         if(count!=10)
         throw new NumberException();
      }
      catch(NumberException e)
      {
         System.out.println(e);
      }
   }
}
```

```
class NumberException extends Exception
{
   public String toString()
   {
      return("Invalid Phone number");
   }
}
```

Output:

```
Enter Phone Number:984172845
Invalid Phone number
```


Write a program to show string constructors.

Source Code:

```
class StringConstructor {
   public static void main(String[] args) {
      char chars[]={'a','b','c'};
      String s=new String(chars);
      String s1=new String(chars,1,2);
      char c[]={'j','a','v','a'};
      String s2=new String(c);
      String s3=new String(s2);
      byte ascii[]={65,66,67,68,69};
      String s4=new String(ascii);
      String s5=new String(ascii,2,3);
      System.out.println(s);
      System.out.println(s1);
      System.out.println(s2);
      System.out.println(s3);
      System.out.println(s4);
      System.out.println(s5);
      System.out.println(s.length());
   }
}
```

Output:

```
abc
bc
java
java
ABCDE
CDE
3
```

Write a program to show demo of concat.

Source Code:

```
class StringConcatDemo {
    public static void main(String[] args) {
        // Create two strings
        String str1 = "Ashesh ";
        String str2 = "Neupane";
        // Concatenate the strings using concat() method
        String result = str1.concat(str2);
        System.out.println("Concatenated String: " + result);
    }
}
```

Output:

```
Concatenated String: Ashesh Neupane
```


Write a program to show demo of toString.

Source Code:

```
class A {
   public String toString() {
      return "Ashesh Neupane";
   }
}
public class toStringDemo {
   public static void main(String[] args) {
      A ob=new A();
      System.out.println(ob);
   }
}
```

Output:

```
Ashesh Neupane
```


Consider a String array:
String phonelist[] = { "+9779861123456" , "9779861123457" , "+977-9849987654" "9849656321"};
Print Only last 10 digit as phone number from the above array.

Source Code:

```
class Phone {
   public static void main(String[] args) {
      String phonelist[] = {"+9779861123456","9779861123457","+977-9849987654","9849656321"};
      for(String v:phonelist)
      {
         System.out.println(v.substring(v.length()-10));
      }
   }
}
```

Output:

```
9861123456
9861123457
9849987654
9849656321
```


Write a program which contains a String countries[] = {"nepal", "india", "china", "australia"};. Print only those countries from the array which starts and ends with same character.

Source Code:

```
class A {
   public static void main(String[] args) {
      String countries[] = {"nepal","india","china","australia"};
      System.out.println("Name of country starting and ending with same character: ");
      for(int i=0;i<countries.length;i++)
      {
         String s=countries[i];
         char start=s.charAt(0);
         char end=s.charAt(s.length()-1);
         if(start==end)
         System.out.println(s);
      }
   }
}
```

Output:

```
Name of country starting and ending with same character:
australia
```


Write a program to print all the vowels present in String object.

Source Code:

```
class A {
   public static void main(String[] args) {
      String s="Ashesh";
      char c[]=s.toCharArray();
      System.out.print("Vowels present: ");
      for(int i=0;i<c.length;i++)
      {
         if(c[i]=='a'||c[i]=='e'||c[i]=='i'||c[i]=='o'||c[i]=='u'||c[i]=='A'||c[i]=='E'||c[i]=='I'||c[i]=='O'||c[i]=='U')
         System.out.print(c[i]+" ");
      }
   }
}
```

Output:

```
Vowels present: A e
```


Write a program to sort the string array that contains name of five students.

Source Code:

```
class StudentSort {
   public static void main(String[] args) {
      String a[]={"bishow","ashesh","sanyam","dipeen","nikesh"};
      System.out.println("Result after sorting:");
      for(int i=0;i<a.length;i++)
      {
         for(int j=i+1;j<a.length;j++)
         {
            if(a[j].compareTo(a[i])<0)
            {
               String temp=a[i];
               a[i]=a[j];
               a[j]=temp;
            }
         }
         System.out.print(a[i]+" ");
      }
   }
}
```

Output:

```
Result after sorting:
ashesh bishow dipeen nikesh sanyam
```


Write a program to demonstrate use of Boolean wrapper class.

Source Code:

```
class Test
{
	public static void main(String[] args)
	{
		boolean b1 = Boolean.parseBoolean("True");
		boolean b2 = Boolean.parseBoolean("TruE");
		boolean b3 = Boolean.parseBoolean("False");
		boolean b4 = Boolean.parseBoolean("FALSE");
		boolean b5 = Boolean.parseBoolean("HighApproach");
		System.out.println(b1);
		System.out.println(b2);
		System.out.println(b3);
		System.out.println(b4);
		System.out.println(b5);
	}
}
```

Output:

```
true
true
false
false
false
```


Write a program to create file and folder in java.

Source Code:

```
import java.io.*;
public class FileDemo {
   public static void createDir() {
      File f = new File("E:\\Ashesh\\2080");
      if (!f.exists()) {
         boolean folderCreated = f.mkdirs();
         if (folderCreated) {
            System.out.println("Directory Created");
         } else {
            System.out.println("Directory Creation Failed!");
         }
      }
   }
   public static void createFile() {
      File f = new File("E:\\Ashesh\\2080\\labreport.txt");
      if (!f.exists()) {
         try {
            if (f.createNewFile()) {
               System.out.println("File Created");
            } else {
               System.out.println("File Creation Failed!");
            }
         } catch (IOException e) {
            System.out.println(e);
         }
      }
   }
   public static void main(String[] args) {
      createDir();
```

```
        createFile();
    }
}
```

Output:

```
Directory Created
File Created
```


Write a program that writes in a text file using filewriter.

Source Code:

```
import java.io.*;
public class writestring {
   public static void main(String[] args) {
      try {
         FileWriter fw=new FileWriter("E:\\Ashesh\\2080\\labreport.txt");
         fw.write('a');
         fw.write("KP Oli");
         char c[]={'a','b','c'};
         fw.write(c);
         fw.close();
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
}
```

Output:

```
aKP Oliabc
```


Write a program that reads the text in a file using Filereader()

Source Code:

```
import java.io.*;
public class Filereader {
   public static void main(String[] args) {
      try {
         FileReader fr=new FileReader("E:\\Ashesh\\2080\\labreport.txt");
         int c;
         while((c=fr.read())!=-1) {
            System.out.print((char)c);
         }
         fr.close();
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
}
```

Output:

```
aKP Oliabc
```


Write a program that writes in a text file using bytestream.

Source Code:

```
import java.io.*;
public class classreading {
   public static void main(String[] args) {
      try {
         FileOutputStream fos=new FileOutputStream("E:\\Ashesh\\2080\\labreport.txt");
         fos.write(65);
         byte b[]={115,104,101,115,104};
         fos.write(b);
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
}
```

Output:

```
Ashesh
```


Write a program that read text file using bytestream.

Source Code:

```
import java.io.*;
public class readstring {
   public static void main(String[] args) {
      try {
         int c;
         FileInputStream fis=new FileInputStream("E:\\Ashesh\\2080\\labreport.txt");
         while((c=fis.read())!=-1) {
            System.out.print((char)c);
         }
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
}
```

Output:

```
Ashesh
```

Write a program for writing object in the file by implementing the Serializable interface.

Source Code:

```
import java.io.*;
class student implements Serializable {
   String name,address;
   int rollno;
   public student(String n, String a, int r) {
      this.name=n;
      this.address=a;
      this.rollno=r;
   }
   public void printinfo() {
      System.out.println("Name:"+this.name);
      System.out.println("Address:"+this.address);
      System.out.println("Roll No:"+this.rollno);
   }
}
public class writingobjectdemo {
   public static void main(String[] args) {
      try {
         student s=new student("Ashesh","Baneshwor",9);
         FileOutputStream fos=new FileOutputStream("E:\\Ashesh\\2080\\labreport.txt");
         ObjectOutputStream oos=new ObjectOutputStream(fos);
         oos.writeObject(s);
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
```

```
}
```


Write a program for reading object of the file by implementing the Serializable interface.

Source Code:

```
import java.io.*;
class student implements Serializable {
   String name,address;
   int rollno;
   public student(String n, String a, int r) {
      this.name=n;
      this.address=a;
      this.rollno=r;
   }
   public void printinfo() {
      System.out.println("Name:"+this.name);
      System.out.println("Address:"+this.address);
      System.out.println("Roll No:"+this.rollno);
   }
}
public class readingobjectdemo {
   public static void main(String[] args) {
      try {
         FileInputStream fis=new FileInputStream("E:\\Ashesh\\2080\\labreport.txt");
         ObjectInputStream ois=new ObjectInputStream(fis);
         Student s=(student)ois.readobject();
         s.printinfo();
      }
      catch(Exception e)
      {
         System.out.println(e);
      }
   }
```

```
}
```

Output:

```
Name: Ashesh
Address: Baneshwor
Roll No: 9
```


Write a program to demonstrate the concept of autoboxing & unboxing.

Source Code:

```
class Boxing {
    public static void main(String[] args) {
        int a=50;
        Integer a1=new Integer(a);
        Integer a3=a;
    }
}
class Demo {
    public static void main(String[] args) {
        Integer i=50;
        int a=i; //unboxing
        System.out.print(a);
    }
}
```

Output:

```
50
```


Write a program to demonstrate example of generic class.

Source Code:

```
class Test<T> {
   T a,b;
   public void printData() {
      System.out.println(a+" "+b);
   }
}
public class Generics {
   public static void main(String[] args) {
      Test<String> t = new Test<>();
      t.a="Hello";
      t.b="Ashesh";
      t.printData();
      Test<Integer>t1=new Test<>();
      t1.a=1;
      t1.b=2;
      t1.printData();
   }
}
```

Output:

```
Hello Ashesh
1 2
```


Write a program to demonstrate generic constuctor.

Source Code:

```
class GenConst {
   public <T> GenConst(T a, T b) {
      System.out.println(a);
      System.out.println(b);
   }
}
public class Ashesh {
   public static void main(String[] args) {
      GenConst g=new GenConst(5,10);
   }
}
```

Output:

```
5
10
```


Write a program to demonstrate generic interface.

Source Code:

```
interface Example<T> {
    void add(T t);
}
class GenInterface implements Example<Integer> {
    int t;
    public void add(Integer t) {
        this.t=t;
    }
    public int get() {
        return t;
    }
}
public class Ashesh {
    public static void main(String[] args) {
        GenInterface ob=new GenInterface();
        ob.add(10);
        System.out.println(ob.get());
    }
}
```

Output:

```
10
```

****** The End ******

Best Of Luck for Semester Examination