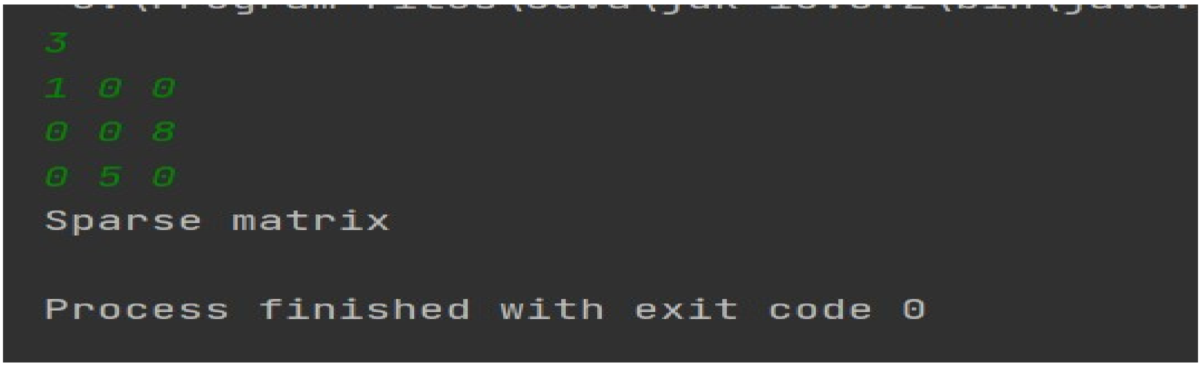
OOPS LAB ASSIGNMENT 1

1.Write a program in Java to determine whether a given matrix is a sparse matrix or not

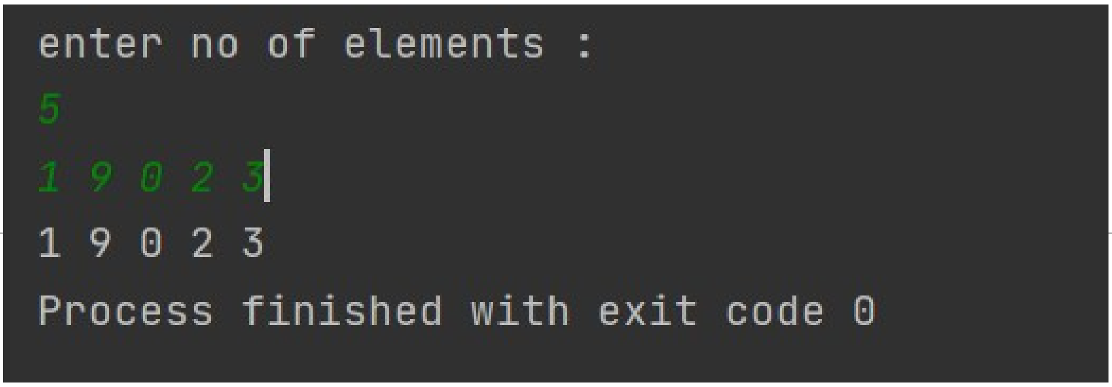
```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(), c = 0;
        int a[][];
        a = new int[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                a[i][j] = sc.nextInt();
                if (a[i][j] == 0)
                    c++;
            }
        }
        if (c > n * n / 2)
            System.out.println("Sparse matrix");
        else
            System.out.println("Not Sparse matrix");
    }
}
```



```
3
1 0 0
0 0 8
0 5 0
Sparse matrix

Process finished with exit code 0
```

2. Write a Java Program to create and display a singly linked list.

```
package com.deepak;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        int n, t=0;
        System.out.println("enter no of elements : ");
        n = sc.nextInt();
        list a;
        a = new list();
        while (n>0) {
            t = sc.nextInt();
            a.ins(t);
            n--;
        }
        a.display();
    }
}
class lnode
{
    int data;
    lnode next;

    lnode(int i)
    {
        data=i;
        next=null;

    }
}
class list {
    lnode head;

    list() {
        head = null;
    }

    void ins(int x) {
        if (head == null) {
            head = new lnode(x);
            return;
        }
        lnode curr = head;
        while (curr.next != null) {
            curr = curr.next;
        }
        curr.next = new lnode(x);
        return;
    }

    void display() {
        lnode curr = head;
        while (curr != null) {
            System.out.print(curr.data + " ");
            curr = curr.next;
        }
    }
}
```

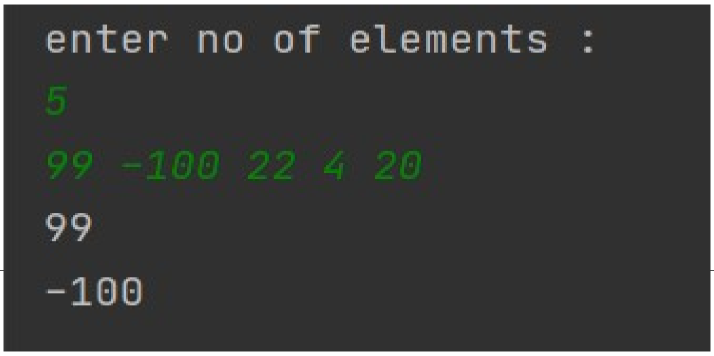


```
enter no of elements :
5
1 9 0 2 3
1 9 0 2 3
Process finished with exit code 0
```

3.Write a Java program to find the maximum and minimum value node from a linked list

```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        int n, t=0;
        System.out.println("enter no of elements : ");
        n = sc.nextInt();
        list a;
        a = new list();
        while (n>0) {
            t = sc.nextInt();
            a.ins(t);
            n--;
        }
        System.out.println(a.max_el());
        System.out.println(a.min_el());
    }
}
class lnode
{
    int data;
    lnode next;

    lnode(int i)
    {
        data=i;
        next=null;

    }
}
class list
{
    lnode head;
    list()
    {
        head=null;
    }
    void ins(int x)
    {
        if (head==null)
        {
            head=new lnode(x);
            return;
        }
        lnode curr=head;
        while (curr.next!=null)
        {
            curr=curr.next;
        }
        curr.next=new lnode(x);
        return;
    }

    void display()
    {
        lnode curr=head;
        while (curr!=null)
        {
            System.out.print(curr.data+" ");
            curr=curr.next;
        }
    }
    int max_el()
    {
        if (head==null)
        {
            System.out.println("empty list ");
            return -1;
        }
        int j=head.data;
        lnode curr=head;
        while (curr!=null)
        {
            if (curr.data>j)
                j=curr.data;
            curr=curr.next;
        }
        return j;
    }
    int min_el()
    {
        if (head==null)
        {
            System.out.println("empty list ");
            return -1;
        }
        int j=head.data;
        lnode curr=head;
        while (curr!=null)
        {
            if (curr.data<j)
                j=curr.data;
            curr=curr.next;
        }
        return j;
    }
}
```

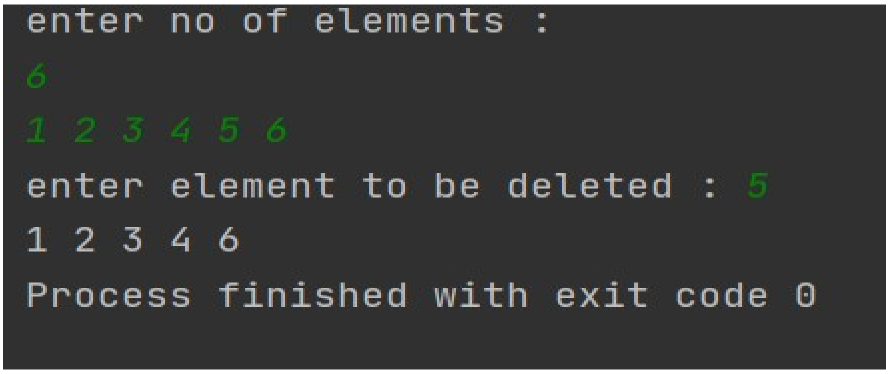
first max is displayed then min element.



```
enter no of elements :
5
99 -100 22 4 20
99
-100
```

4. Write a Java program to delete a node from the middle of the singly linked list

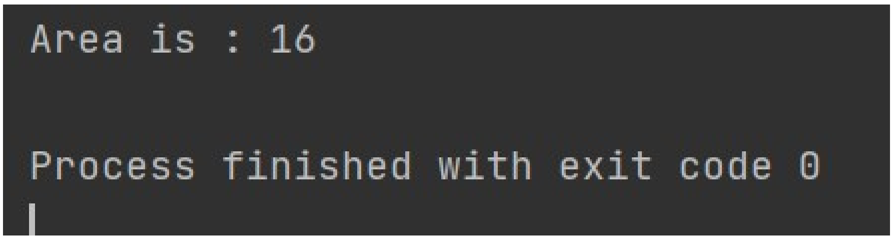
```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        int n, t=0;
        System.out.println("enter no of elements : ");
        n = sc.nextInt();
        list a;
        a = new list();
        while (n>0) {
            t = sc.nextInt();
            a.ins(t);
            n--;
        }
        System.out.print("enter element to be deleted : ");
        t= sc.nextInt();
        a.del(t);
        a.display();

    }
}
class lnode
{
    int data;
    lnode next;

    lnode(int i)
    {
        data=i;
        next=null;
    }
}
class list
{
    lnode head;
    list()
    {
        head=null;
    }
    void ins(int x)
    {
        if (head==null)
        {
            head=new lnode(x);
            return;
        }
        lnode curr=head;
        while (curr.next!=null)
        {
            curr=curr.next;
        }
        curr.next=new lnode(x);
        return;
    }

    void display()
    {
        lnode curr=head;
        while (curr!=null)
        {
            System.out.print(curr.data+" ");
            curr=curr.next;
        }
    }
    void del(int x)
    {
        if (head==null)
            return;
        if (head.data==x)
        {
            head=head.next;
            return;
        }
        lnode curr=head,prev=head;
        while (curr.data!=x)
        {
            prev=curr;
            curr=curr.next;
        }
        prev.next=curr.next;
    }
}
```



```
enter no of elements :
6
1 2 3 4 5 6
enter element to be deleted : 5
1 2 3 4 6
Process finished with exit code 0
```

5.Write a program in Java to implement multilevel inheritance which shows the usage of super and final keyword

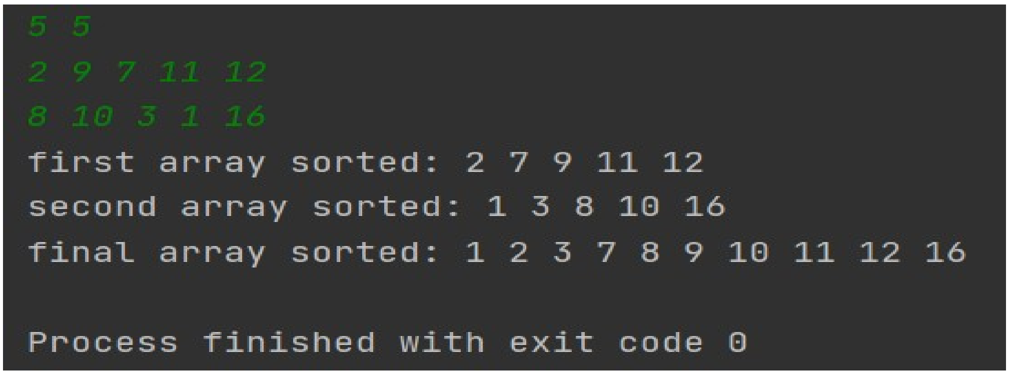
```
package com.deepak;
import java.util.Scanner;

class figure {
    int dim1,dim2;

    figure(int a, int b)
    {
        dim1=a;
        dim2=b;
    }
    //final method implies its subclasses cannot override this method
    final void area() {
        System.out.println("Area is : "+ dim1*dim2);
    }
}
class rectangle extends figure {

    rectangle(int x, int y) {
        super(x, y);
    }
}
// final states that this class cannot be further inherited
final class square extends rectangle {
    square(int a, int b) {
        super(a, b);
    }
}

public class Main {

    public static void main(String[] args) {
        square a= new square(4,4);
        a.area();
    }
}
```



```
Area is : 16

Process finished with exit code 0
```

6.Write a java program to create two arrays(Unsorted) then sort them by using the best sorting algorithm (Recommended quick sort) then merge these two arrays.

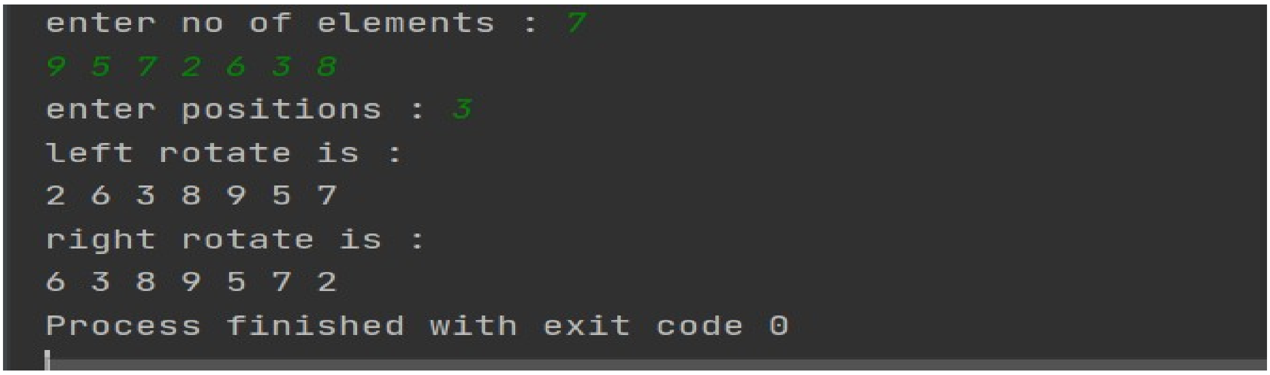
```
package com.deepak;
import java.util.Scanner;


public class Main {
    static int partion(int[] arr, int low, int high) {
        int pivort = arr[high];
        int i = low - 1;
        for (int j = low; j < high; j++) {
            if (arr[j] <= pivort) {
                i += 1;
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
        int temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
        return i + 1;
    }
    static void quicksort(int[] arr, int low, int high) {
        if (low < high) {
            int p = partion(arr, low, high);
            quicksort(arr, low, p - 1);
            quicksort(arr, p + 1, high);
        }
    }
    static void print(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int size[] = new int[2];
        size[0] = sc.nextInt();
        size[1] = sc.nextInt();
        int arr1[] = new int[size[0]];
        int arr2[] = new int[size[1]];
        for (int i = 0; i < arr1.length; i++) {
            arr1[i] = sc.nextInt();
        }
        for (int i = 0; i < arr2.length; i++) {
            arr2[i] = sc.nextInt();
        }
        quicksort(arr1, 0, arr1.length - 1);
        quicksort(arr2, 0, arr2.length - 1);
        System.out.print("first array sorted: ");
        print(arr1);
        System.out.print("second array sorted: ");
        print(arr2);
        int[] arr = new int[size[0] + size[1]];
        int index1 = 0, index2 = 0;
        for (int i = 0; i < arr.length; i++) {
            if (index1 == arr1.length) {
                arr[i] = arr2[index2];
                index2 += 1;
                continue;
            }
            if (index2 == arr2.length) {
                arr[i] = arr1[index1];
                index1 += 1;
                continue;
            }
            if (arr1[index1] < arr2[index2]) {
                arr[i] = arr1[index1];
                index1 += 1;
            } else {
                arr[i] = arr2[index2];
                index2 += 1;
            }
        }
        System.out.print("final array sorted: ");

        print(arr);

        sc.close();
    }
}
```



```
5 5
2 9 7 11 12
8 10 3 1 16
first array sorted: 2 7 9 11 12
second array sorted: 1 3 8 10 16
final array sorted: 1 2 3 7 8 9 10 11 12 16

Process finished with exit code 0
```

7.Write a java program to left and right rotate the array by a given number of positions. (Number of positions to shift must be read from the console.)

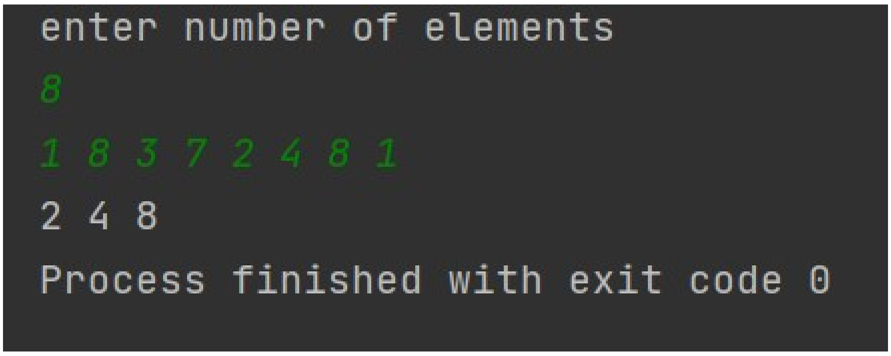
```java
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        int n,k=0;
        System.out.print("enter no of elements : ");
        n=sc.nextInt();
        int a[],temp[];
        a=new int[n];
        temp=new int[n];
        for (int i=0;i<n;i++)
            a[i]=sc.nextInt();
        System.out.print("enter positions : ");
        k=sc.nextInt();
        k=k%n;
        System.out.println("left rotate is : ");
        for (int i=0;i<n;i++)
            temp[(i-k+n)%n]=a[i];
        for (int i=0;i<n;i++)
            System.out.print(temp[i]+" ");
        System.out.println();
        System.out.println("right rotate is : ");
        for (int i=0;i<n;i++)
            temp[(i+k)%n]=a[i];
        for (int i=0;i<n;i++)
            System.out.print(temp[i]+" ");
    }
}
```



```
enter no of elements : 7
9 5 7 2 6 3 8
enter positions : 3
left rotate is :
2 6 3 8 9 5 7
right rotate is :
6 3 8 9 5 7 2
Process finished with exit code 0
```

8.Given an unsorted array of elements, find the longest consecutive elements sequence (must be in ascending order) in the array using a Java program.

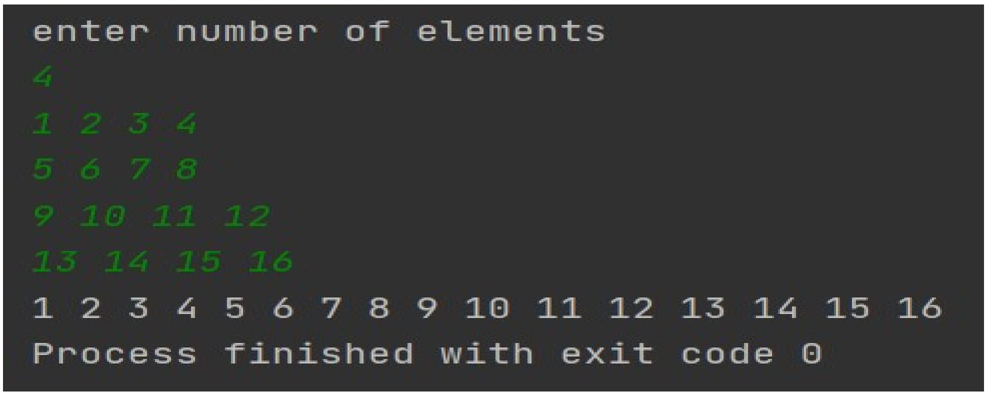
```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        System.out.println("enter number of elements ");
        int n = sc.nextInt(),k=0, c = 0,ind=-1,a[];
        a = new int[n];
        for (int i=0;i<n;i++)
            a[i]=sc.nextInt();
        int j=0,t=0;
        while (j<n)
        {
            t=0;
            int i=j;
            while ((j+1<n)&&(a[j]<a[j+1]))
            {j++;
            t++;}
            if (t>c)
            {
                c=t;
                k=i;
            }
            j++;
        }
        c+=1;
        for (int i=k;i<k+c;i++)
            System.out.print(a[i]+ " ");
    }
}
```



```
enter number of elements
8
1 8 3 7 2 4 8 1
2 4 8
Process finished with exit code 0
```

9.Write a java program to create a 2D array and write code to print the spiral traversal of that array? Take array length and read array from the console.

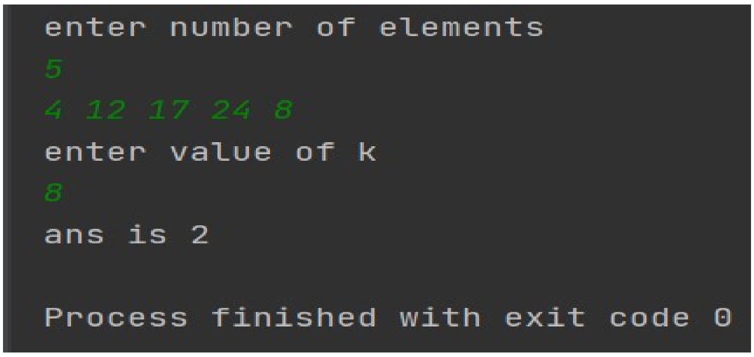
```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        System.out.println("enter number of elements ");
        int n = sc.nextInt(),k=0, c = 0,ind=-1,a[][];
        a = new int[n][n];
        for (int i=0;i<n;i++)
        {
            for (int j=0;j<n;j++)
                a[i][j]=sc.nextInt();
        }
        for (int i=0;i<n;i++)
        {
            for (int j=0;j<n;j++)
                System.out.print(a[i][j]+ " ");
        }

    }
}
```



```
enter number of elements
4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Process finished with exit code 0
```

10.Given an array, write a java program to find an index of the smallest element such that array elements sum will be divisible by k.

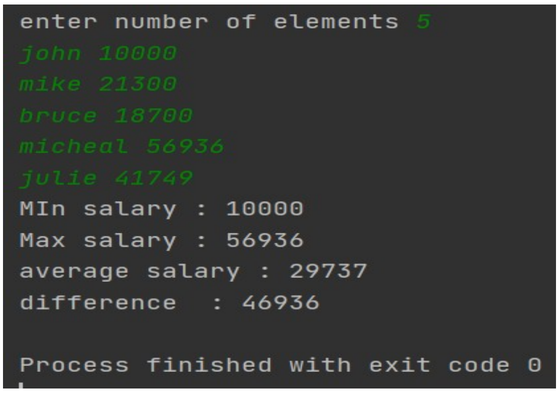
```
package com.deepak;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // write your code here
        Scanner sc = new Scanner(System.in);
        System.out.println("enter number of elements ");
        int n = sc.nextInt(),k, c = 0,ind=-1,a[];
        a = new int[n];
        for (int i = 0; i < n; i++) {
          a[i]=sc.nextInt();
        }
        System.out.println("enter value of k ");
        k=sc.nextInt();
        for (int i=0;i<n;i++)
        {
          c+=a[i];
          if (c%k==0)
          {
            ind=1+i;
            break;
          }
        }
        System.out.println("ans is "+ ind);
    }
}
```



```
enter number of elements
5
4 12 17 24 8
enter value of k
8
ans is 2

Process finished with exit code 0
```

11.Create a class called as *Company* which is a base class with *name* and *salary* as fields and take 5 employee details as input from the user. Derive one class containing methods *dispMin()* and *dispMax()* to display the employee name with minimum and maximum salary. Derive another class containing the methods *avgSalary()* and *difference()* which displays the average salary of employee and the difference between maximum and minimum salary. Write a Java program to implement the same.

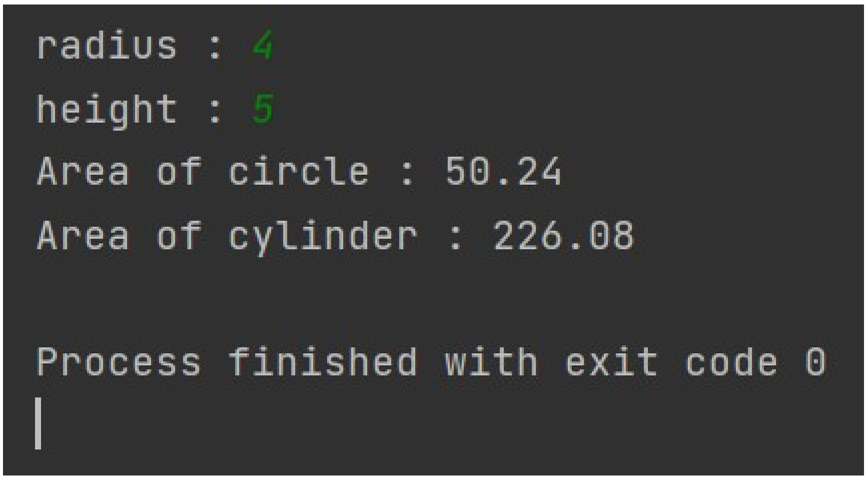
```
package com.deepak;
import java.util.Scanner;

class company{
    int salary[];
    String name[];
    Scanner sc = new Scanner(System.in);
     company(int i)
     {
        salary=new int[i];
        name=new String[i];
        for (int j=0;j<i;j++)
        {
          name[j]=sc.next();
          salary[j]=sc.nextInt();
        }
     }
}

class dis extends company
{
    dis(int i) {
        super(i);
    }

    int dispMin()
    {
        int m=salary[0];
        for (int i=0;i<salary.length;i++)
            m=Math.min(m,salary[i]);
        //System.out.println(m);
        return m;
    }

    int dispMax()
    {
        int m=salary[0];
        for (int i=0;i<salary.length;i++)
            m=Math.max(m,salary[i]);
       // System.out.println(m);
        return m;
    }

}

class another extends dis
{

    another(int i) {
        super(i);
    }

    int avg()
    {
        int m=0;
        for (int i=0;i<salary.length;i++)
            m+=salary[i];
        return m/salary.length;
    }

    int difference()
    {
        return super.dispMax()-super.dispMin();
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("enter number of elements ");
        another a=new another(sc.nextInt());
        System.out.println("MIn salary : "+ a.dispMin());
        System.out.println("Max salary : "+ a.dispMax());
        System.out.println("average salary : "+ a.avg());
        System.out.println("difference  : "+ a.difference());
    }
}
```



```
enter number of elements 5
john 10000
mike 21300
bruce 18700
micheal 56936
julie 41749
MIn salary : 10000
Max salary : 56936
average salary : 29737
difference  : 46936

Process finished with exit code 0
```

12.Create a base class *Shape* containing *name* as field. Class *Shape* will have a public method called getName() that returns the name of the shape. Create a class *Circle* deriving *Shape* having *radius* as field and *calculate()* method to calculate the area. Then, create a class *Cylinder* deriving *Circle* having *height* as field and *calculate()* method. Take the radius and height values from user as input and calculate areas of circle and cylinder as output. Write a Java program to implement the same

```java
package com.deepak;
import java.util.Scanner;

class shape
{
    String s;
    shape(String a)
    {
        s=a;
    }

    public String getName()
    {
        return s;
    }
}
class circle extends shape
{
    double radius;
    circle(int r) {
        super("circle");
        radius=r;
    }
    double calculate()
    {
        return 3.14*radius*radius;
    }
}

class cylinder extends circle
{
    int height;
    cylinder(int r,int h ){
        super(r);
        height=h;
    }
    double circle_area()
    {
        return super.calculate();
    }
    double calculate()
    {
        return 2*super.calculate()+2*3.14*radius*height;
    }

}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int r,h;
        System.out.print("radius : ");
        r=sc.nextInt();
        System.out.print("height : ");
        h=sc.nextInt();
        cylinder a=new cylinder(r,h);
        System.out.println("Area of circle : "+ a.circle_area());
        System.out.println("Area of cylinder : "+ a.calculate());
    }
}
```



```
radius : 4
height : 5
Area of circle : 50.24
Area of cylinder : 226.08

Process finished with exit code 0
```