Q1Write Java programs

a. **To print fibonacci series without using recursion and using recursion.(concept of loops, data types)**

```
// fibonacci series without using recursion
import java.util.Scanner;
class fib{
   public static void main(String[] args){


      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the number of terms you want:\t");
      int n = sc.nextInt();


      int a=0,b=1,c;
      int i = 0;
      System.out.print("The fibonacci series is: ");
      while(i<n){
         System.out.print(a + " ");
         c = a+b;
         a = b;
         b = c;
         i++;
      }
   }
}
```

```
//fibonacci series without using recursion
import java.util.Scanner;
class fibrec{
   static int fib(int n){
      if(n == 0){
         return 0;
      }
      else if(n == 1){
         return 1;
      }
      else{
         return (fib(n-1)+fib(n-2));
      }
   }
```

```
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int n;
        System.out.print("Enter the number of terms you want :\t");
        n = sc.nextInt();
        int i=0;
        System.out.print("The fibonacci series is : \t");
        while(i<n){
            System.out.print(fib(i)+" ");
            i++;
        }
    }
}
```

Ouput

```
Enter the number of terms you want: 6
The fibonacci series is: 0 1 1 2 3 5
```

Output

```
Enter the number of terms you want :            11
The fibonacci series is :          0 1 1 2 3 5 8 13 21 34 55
```

Q1 Write Java programs

b. To check prime numbers.

```java
import java.util.Scanner;
class prime{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number you want to check:\t");
        int n = sc.nextInt();
        int flag =0;
        for(int i=2;i<=n/2;i++){
            if(n%i == 0){
                flag = 1;
                break;
            }
            else{
                flag = 0;
            }
        }
        if(flag == 1){
            System.out.println("The entered number is not a prime number");
        }
        else{
            if(n != 1 || n != 0){
                System.out.println("The entered number is a prime number");
            }
            else{
                System.out.println("The entered number is neither prime nor composite");
            }
        }
    }
}
```

Output

```
Enter the number you want to check: 3
The entered number is a prime number
```

Q1. Write Java programs

c. To sort an array elements using bubble sort algorithm.

```
import java.util.Scanner;
class bubbleSort{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elements in the array");
        int n = sc.nextInt();
        int[] array = new int[n];
        System.out.println("Enter the values");
        for(int i=0;i<n;i++){
            array[i] = sc.nextInt();
        }
        for(int i=0;i<n;i++){
            for(int j=0;j<n-i-1;j++){
                if(array[j]>array[j+1]){
                    int temp = array[j+1];
                    array[j+1] = array[j];
                    array[j] = temp;
                }
            }
        }
        System.out.print("The sorted array is:\t");
        for(int i=0;i<n;i++){
            System.out.print(array[i]+" ");
        }
    }
}
```

Output

```
Enter the number of elements in the array
5
Enter the values
2
1
7
3
5
The sorted array is:    1 2 3 5 7
```

Q2. Create a class called account with the data members(Accno – integer, name String, Phone_No: integer, balance_amt:float), and following methods :

a. **getinput() to get input from the user**
b. **Deposit() method which takes the amount to be deposited in to his/her account and do the calculation.**
c. **Withdraw() method which gets the amount to be withdrawn from his/her account.**
d. **Print the appropriate results.**

```
import java.util.Scanner;
class account{
    Scanner sc = new Scanner(System.in);
    long accNo;
    long phoneNo;
    String name;
    double balanceAmt;
    public void getInput(){
        System.out.print("Enter the account number:\t");
        accNo = sc.nextLong();
        sc.nextLine();
        System.out.print("Enter the account holders name:\t");
        name = sc.nextLine();
        System.out.print("Enter the account holders phone number:\t");
        phoneNo = sc.nextLong();
         balanceAmt =0;
    }
    public void deposit(){
        System.out.println("Enter the amount you wanna deposit:\t");
        double deposit = sc.nextDouble();
        balanceAmt += deposit;
        System.out.println("Amount Deposited Successfully");
        System.out.println("The account balance is:\t"+balanceAmt);
    }
    public void withdraw(){
        System.out.println("Enter the amount you wanna withdraw:\t");
        double withdraw = sc.nextDouble();
        if(balanceAmt-withdraw>=0){
            balanceAmt -= withdraw;
            System.out.println("Amount withdrawn successful");
        }
        else{
            System.out.println("Your account has insufficient funds");
        }
        System.out.println("The account balance is:\t"+balanceAmt);
    }
    public void printDetails(){
        System.out.println("\nAccount Number:\t"+accNo);
        System.out.println("\nAccount holders name:\t"+name);
        System.out.println("\nPhone number:\t"+phoneNo);
        System.out.println("\nCurrent balance in the account:\t"+balanceAmt);
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        account a1 = new account();
    while(true){
        System.out.println("1.Enter the user details");
        System.out.println("2.Deposit money in the account");
        System.out.println("3.Withdraw money from the account");
        System.out.println("4.Get the account details");
        System.out.println("5. Exit!");
        System.out.print("Enter your choice:\t");
        int ch = sc.nextInt();
        switch(ch){
            case 1:
                    a1.getInput();
                    break;
            case 2:
                    a1.deposit();
                    break;
            case 3:
                    a1.withdraw();
                    break;
            case 4:
                    a1.printDetails();
                    break;
            case 5:
                    System.exit(0);
                    break;
            default:
                    System.out.println("Invalid choice!");
        }
        System.out.println("");
    }
    }
}
```

Output

```
1.Enter the user details2.Deposit money in the account
3.Withdraw money from the account
4.Get the account details
5. Exit!
Enter your choice:      1
Enter the account number:     2132327
Enter the account holders name:       Bharat
Enter the account holders phone number:     7452818390

1.Enter the user details
2.Deposit money in the account
3.Withdraw money from the account
4.Get the account details
5. Exit!
Enter your choice:      2
Enter the amount you wanna deposit:
5000
Amount Deposited Successfully
The account balance is:         5000.0

1.Enter the user details
2.Deposit money in the account
3.Withdraw money from the account
4.Get the account details
5. Exit!
Enter your choice:      3
Enter the amount you wanna withdraw:        100
Amount withdrawn successful
The account balance is:         4900.0

1.Enter the user details
2.Deposit money in the account
3.Withdraw money from the account
4.Get the account details
5. Exit!
Enter your choice:      4
Account Number:       2132327
Account holders name:         Bharat
Phone number:         7452818390
Current balance in the account:         4900.0
```

```
1.Enter the user details
2.Deposit money in the account
3.Withdraw money from the account
4.Get the account details
5. Exit!
Enter your choice:      5
```

Q3.Define a Stack class to implement the stack data structure. Include constructors to perform initialization, method push to push an element into the stack, method pop to remove an element from the stack and display method to display the elements of the stack.

```
import java.util.Scanner;
class stack{
    int top;
    int size;
    int stack[];
    stack(int size,int top){
        this.size = size;
        this.top = top;
        this.stack = new int[size];
    }
    public void push(int ele){
        if(top == size-1){
            System.out.println("Stack Overflow");
        }
        else{
            stack[++top] = ele;
            System.out.println("Element successfully inserted");
        }
    }
    public int pop(){
        if(top == -1){
            return -1;
        }
        else{
            return stack[top--];
        }
    }
    public void display(){
        if(top == -1){
            System.out.println("The stack is empty");
        }
        else{
            System.out.println("The elemnets of the stack are:\t");
            for(int i=top;i>=0;i--){
                System.out.print(stack[i]+" ");
            }
        }
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of the stack:\t");
        int size = sc.nextInt();
        stack s1 = new stack(size,-1);
        System.out.println("1. Push");
        System.out.println("2. Pop");
        System.out.println("3. Display");
        System.out.println("4. Exit!");
        while(true){
            System.out.print("\nEnter your choice:\t");
            int ch = sc.nextInt();
            switch(ch){
                case 1:
                        System.out.print("Enter the element you want to insert:\t");
                        int ele1 = sc.nextInt();
                        s1.push(ele1);
                        break;
                case 2:
                        int ele2 = s1.pop();
                        if(ele2 != -1){
                            System.out.println("The elemnet popped out from the stck is:"+ele2);
                        }
                        else{
                            System.out.println("Stack underflow!");
                        }
                        break;
                case 3:
                        s1.display();
                        break;
                case 4:
                        System.exit(0);
                        break;
                default:
                        System.out.println("Invalid choice!");
            }
        }
    }
}
```

Output

```
Enter the size of the stack:     5
1. Push
2. Pop
3. Display
4. Exit!

Enter your choice:      1
Enter the element you want to insert:3
Element successfully inserted

Enter your choice:      1
Enter the element you want to insert:2
Element successfully inserted

Enter your choice:      1
Enter the element you want to insert:4
Element successfully inserted

Enter your choice:      1
Enter the element you want to insert:3
Element successfully inserted

Enter your choice:      1
Enter the element you want to insert:8
Element successfully inserted

Enter your choice:      1
Enter the element you want to insert:3
Stack Overflow

Enter your choice:      2
The elemnet popped out from the stck is:8

Enter your choice:      2
The elemnet popped out from the stck is:3

Enter your choice:      3
The elemnets of the stack are:
4 2 3
Enter your choice:      4
```

Q4. Define a class Complex with data members as two real numbers, constructors for initialization these numbers, methods to add, subtract and multiply 2 complex numbers.

```
import java.util.Scanner;
class complex{
    Scanner sc = new Scanner(System.in);
    int a;
    int b;
    complex(){
        this.a = 0;
        this.b =0;
    }
    public void getInput(){
        System.out.println("Enter the real part of the complex number");
        a = sc.nextInt();
        System.out.println("Enter the imaginary part of the complex number");
        b = sc.nextInt();
    }
    public complex sum_c(complex c1,complex c2){
        complex c3 = new complex();
        c3.a = c1.a + c2.a;
        c3.b = c1.b + c2.b;
        return c3;
    }
    public complex diff_c(complex c1,complex c2){
        complex c3 = new complex();
        c3.a = c1.a - c2.a;
        c3.b = c1.b - c2.b;
        return c3;
    }
    public complex mul_c(complex c1, complex c2){
        complex c3 = new complex();
        c3.a = (c1.a * c2.a) - (c1.b * c2.b);
        c3.b = (c1.a * c2.b) + (c2.a * c1.b);
        return c3;
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("1. Addition of two complex numbers");
        System.out.println("2. Subtraction of two complex numbers");
        System.out.println("3. Multiplication of two complex numbers");
        System.out.println("4. Exit!");
        complex c1 = new complex();
        complex c2 = new complex();
        complex compute = new complex();
        while(true){
          System.out.println("Enter your choice");
          int ch = sc.nextInt();
          switch(ch){
            case 1:
                System.out.println("Enter the first complex number");
                c1.getInput();
                System.out.println("Enter the second complex number");
                c2.getInput();
                compute = compute.sum_c(c1,c2);
                System.out.println("The sum of "+c1.a + " +i " + c1.b + " and "+c2.a + " +i
" + c2.b +" is "+compute.a + " +i (" + compute.b+")");
                break;
            case 2:
                System.out.println("Enter the first complex number");
                c1.getInput();
                System.out.println("Enter the second complex number");
                c2.getInput();
                compute = compute.diff_c(c1,c2);
                System.out.println("The sum of "+c1.a + " +i " + c1.b + " and "+c2.a + " +i
" + c2.b +" is "+compute.a + " +i (" + compute.b +")" );
                break;
            case 3:
                System.out.println("Enter the first complex number");

                c1.getInput();
                System.out.println("Enter the second complex number");
                c2.getInput();
                compute = compute.mul_c(c1,c2);
                System.out.println("The sum of "+c1.a + " +i " + c1.b + " and "+c2.a + " +i
" + c2.b +" is "+compute.a + " +i (" + compute.b+")"  );
                break;
            case 4:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice");
```

```
                        break;
            }
        }
    }
}
```

Output

```
java -cp /tmp/jyYUfJoDGd complex
1. Addition of two complex numbers
2. Subtraction of two complex numbers
3. Multiplication of two complex numbers
4. Exit!

Enter your choice
1
Enter the first complex number
Enter the real part of the complex number
2
Enter the imaginary part of the complex number
3
Enter the second complex numberEnter the real part of the complex number
4
Enter the imaginary part of the complex number
1
The sum of 2+i3 and 4+i1 is 6+i (4)

Enter your choice
2
Enter the first complex numberEnter the real part of the complex number
1
Enter the imaginary part of the complex number
2
Enter the second complex number
Enter the real part of the complex number
5
Enter the imaginary part of the complex number
2
The sum of 1+i2 and 5+i2 is -4+i(0)
Enter your choice
3
Enter the first complex number
Enter the real part of the complex number
2
Enter the imaginary part of the complex number
3
Enter the second complex number
Enter the real part of the complex number3
Enter the imaginary part of the complex number
4
The sum of 2+i3 and 3+i4 is -6+i(17)

Enter your choice
4
```

Q5. Write a java program to read 2 matrices and place the product in a third matrix . Use constructors and suitable methods.

```
import java.util.Scanner;
class matrix{
    int rows;
    int columns;
    int array[][];
    Scanner sc = new Scanner(System.in);
    matrix(int rows,int columns){
        this.rows = rows;
        this.columns = columns;
        this.array = new int[rows][columns];
    }
    public void getInput(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<columns;j++){
                array[i][j] = sc.nextInt();
            }
        }
    }
    public int matrixMul(matrix m1,matrix m2){
        int flag =0;
        if(m1.columns != m2.rows){
            flag =1;
            return flag;
        }
        else{
            for(int i=0;i<m1.rows;i++){
                for(int j=0;j<m2.columns;j++){
                    for(int k=0;k<m1.columns;k++){
                        array[i][j] += m1.array[i][k] * m2.array[k][j];
                    }
                }
            }
            return flag;
        }
    }
    public void printMatrix(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<columns;j++){
                System.out.print(array[i][j]+" ");
            }
            System.out.println();
        }
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("For matrix 1");
        System.out.print("Enter the no of rows:\t");
        int rows1 = sc.nextInt();
        System.out.print("Enter the no of columns:\t");
        int columns1 = sc.nextInt();
        matrix m1 = new matrix(rows1,columns1);
        System.out.println("Enter the elements of the matrix");
        m1.getInput();
        System.out.println("For matrix 2");
        System.out.print("Enter the no of rows:\t");
        int rows2 = sc.nextInt();
        System.out.print("Enter the no of columns:\t");
        int columns2 = sc.nextInt();
        matrix m2 = new matrix(rows2,columns2);
        System.out.println("Enter the elements of the matrix");
        m2.getInput();
        matrix m3 = new matrix(m1.rows,m2.columns);
       int flag = m3.matrixMul(m1,m2);
       System.out.println("The First matrix is");
        m1.printMatrix();
        System.out.println("The Second matrix is");
        m2.printMatrix();
        if(flag == 0){
            System.out.println("The multiplication of two matrices is");
            m3.printMatrix();
        }
        else{
            System.out.println("The multiplication of the two matrices is not possible");
        }
    }
}
```

Output

```
For matrix 1
Enter the no of rows:  3
Enter the no of columns:        3
Enter the elements of the matrix
9
2
1
4
2
5
8
7
4
1
For matrix 2
Enter the no of rows:  3
Enter the no of columns:        3
Enter the elements of the matrix
1
2
3
4
5
6
7
8
9
10
The First matrix is
9 2 4
2 5 8
7 4 1
The Second matrix is
1 2 4
5 6 7
8 9 10
The multiplication of two matrices is
51 66 90
91 106 123
35 47 66
```

Q6.Write a java program to work with strings.

a. **Extract a portion of the string and print it. Variable m indicates the amount of characters to be extracted from the string starting from the n^th^ position.**
b. **Read a text and count all the occurrences of a particular word.**
c. **Replace a substring in the given string.**
d. **Rearrange the string and rewrite in alphabetical order.**
e. **Compare two strings ignoring case.**
f. **Concatenate two strings.**

```
import java.util.Scanner;
import java.util.Arrays;
class strings{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      System.out.println("1. Extract a portion of a string");
      System.out.println("2. Count the occurences of a particular word in a string");
      System.out.println("3. Replace a substring in a given string");
      System.out.println("4. Rearrange the strings in alphabetical order");
      System.out.println("5. Compare two strings ignoring case");
      System.out.println("6. Concatenate two strings");
      System.out.println("7. Exit!");
      while(true){
         System.out.print("Enter your choice:\t");
         int ch = sc.nextInt();
         sc.nextLine();
         switch(ch){
            case 1:
                  System.out.print("Enter the String:\t");
                  String s = sc.nextLine();
                  System.out.print("\nEnter the position of the string from which you wanna
start extracting:\t");
                  int n = sc.nextInt();
                  System.out.print("\nEnter the no of characters to be extracted:\t");
                  int m = sc.nextInt();
                  String k = s.substring(n-1,m+n-1);
                  System.out.println("\nThe extracted string is: "+k);
                  break;
            case 2:
                  System.out.print("Enter the String:\t");
                  s = sc.nextLine();
                  System.out.print("\nEnter the word whose occurences you wanna
find:\n");
                  k = sc.next();
                  int counter = 0;
                  String arr[] = s.split(" ");
                  for(int i=0;i<arr.length;i++){
                      if(arr[i].equals(k)){
                          counter++;
                      }
                  }
                  System.out.println("\nThe no of occurences of the word "+k+" are
"+counter);
                  break;
            case 3:
                  System.out.print("Enter the string:\t");
                  s = sc.nextLine();
                  System.out.print("\nEnter the substring you wanna replace:\t");
                  k = sc.next();
                  System.out.print("\nEnter the substring you wanna replace with:\t");
                  String p = sc.next();
                  String t = s.replace(k,p);
                  System.out.println("\nThe new string is: "+t);
                  break;
               case 4:
                  System.out.print("Enter the string:\t");
                  s = sc.nextLine();
                  k = s.toLowerCase();
                  char[] arr2 = k.toCharArray();
                  Arrays.sort(arr2);
                  p = new String(arr2);
                  System.out.println("\nThe string arranged aphabetically  is: "+p);
                  break;
            case 5:
                  System.out.print("Enter the string 1:\t");
                  s = sc.nextLine();
                  System.out.print("\nEnter the string 2:\t");
                  k = sc.nextLine();
                  boolean cmp = s.equalsIgnoreCase(k);
                  if(cmp){
                      System.out.println("\nThe two strings are equal");
                  }
                  else{
                      System.out.println("\nThe two strings are not equal");
                  }
```

```
                        break;
                case 6:
                        System.out.print("Enter the string 1:\t");
                        s = sc.nextLine();
                        System.out.print("Enter the string 2:\t");
                        k = sc.nextLine();
                        p = s+" "+k;
                        System.out.println("\nThe concatenated strings is: "+p);
                        break;
                case 7:
                        System.exit(0);
                        break;
                default:
                        System.out.println("Invalid choice");
            }
        }
    }
}
```

Ouptut

```
1. Extract a portion of a string
2. Count the occurences of a particular word in a string
3. Replace a substring in a given string
4. Rearrange the strings in alphabetical order
5. Compare two strings ignoring case
6. Concatenate two strings
7. Exit!
Enter your choice:      1
Enter the String:       Computer is my favourite subject to study.
Enter the position of the string from which you wanna start extracting:      10
Enter the no of characters to be extracted:   10
The extracted string is: is my favo

Enter your choice:      2
Enter the String:       java
Enter the word whose occurences you wanna find:
a
The no of occurences of the word a are 0

Enter your choice:      3
Enter the string:       Rashi mittal
Enter the substring you wanna replace:      mittal
Enter the substring you wanna replace with: agarwal
The new string is: Rashi agarwal

Enter your choice:      4
Enter the string:       uefhhcqefda
The string arranged aphabetically  is: acdeeffhhqu

Enter your choice:      5
Enter the string 1:     java
Enter the string 2:     java
The two strings are equal

Enter your choice:      6
Enter the string 1:     Object Oriented
Enter the string 2:     Programing Laboratory
The concatenated strings is: Object Oriented Programing Laboratory

Enter your choice:      7
```

Q7.Create a Personal class to hold the personal details of an person such as name, age, education, salary-(0basic, da, hra), years of experience, number of loans and loan amount. Write constructors to assign values to the data members. Include an

a. **isEligible() method to indicate whether the person is eligible for loan,**
b. **taxPay() method to indicate the amount of tax to be paid,**
c. **isEligiblePromotion() to indicate whether the person is eligible for a promotion.**
d. **Display () method to display the details.**

Enter the details of n employees and indicate their eligibility and the tax to be paid.

```
import java.util.Scanner;
class personal{
    String name;
    int age;
    String education;
    double basic;
    double hra;
    double da;
    int years_of_exp;
    int no_of_loans;
    double loan_amt[];
    personal(String name,int age,String education,double basic,double hra,double da,int
years_of_exp,int no_of_loans,double[] loan_amt){
        this.name = name;
        this.age = age;
        this.education = education;
        this.basic = basic;
        this.hra = hra;
        this.da = da;
        this.years_of_exp = years_of_exp;
        this.no_of_loans = no_of_loans;
        this.loan_amt =  new double[no_of_loans];
        this.loan_amt = loan_amt;
    }
    void isEligible(personal p){
        if((p.basic+p.hra+p.da)>500000 && p.no_of_loans <=2){
            System.out.println("The person is eligible for loan");
        }
        else{
            System.out.println("The person is not eleigible for loan");
        }
    }
    void taxPay(personal p){
        double sal = p.basic + p.hra + p.da;
        if(sal<=250000){
            System.out.println("No tax is to be paid");
        }
        else if(sal<=500000){
            System.out.println("The tax to be paid is: "+0.2*sal);
        }
        else if(sal<=1000000){
            System.out.println("The tax to be paid is: "+0.3*sal);
        }
        else{
            System.out.println("The tax to be paid is: "+0.4*sal);
        }
    }
    void isEligiblePromotion(personal p){
        if(p.years_of_exp >=10 && p.age>=43){
            System.out.println("The person is eligible for promotion");
        }
        else{
            System.out.println("The person is not eleigible for promotion");
        }
    }
    void display(personal p){
        System.out.println("Name of the person is: "+p.name);
        System.out.println("Age of the person is: "+p.age);
        System.out.println("Education of the person is: "+p.education);
        System.out.println("The total salary of the person is: "+(p.basic+p.hra+p.da));
        System.out.println("The years of experience of the person is: "+p.years_of_exp);
        System.out.println("The no of loans in the name of the person is: "+p.no_of_loans);
        System.out.print("The loan amount for each loan is:\t ");
            for(int i=0;i<p.no_of_loans;i++){
                System.out.print(p.loan_amt[i]+" ");
            }
             System.out.println();
            p.isEligible(p);
            p.taxPay(p);
            p.isEligiblePromotion(p);
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no of employees whose detail you wanna enter");
        int n = sc.nextInt();
        sc.nextLine();
        personal[] p = new personal[n];
        for(int i=0;i<n;i++){
          System.out.println("Enter the name");
          String name = sc.nextLine();
          System.out.println("Enter the age");
          int age = sc.nextInt();
          sc.nextLine();//Have to give after int
          System.out.println("Enter the education");
          String education = sc.nextLine();
          System.out.println("Enter the basic salary");
          double basic = sc.nextDouble();
          System.out.println("Enter the hra");
          double hra = sc.nextDouble();
          System.out.println("Enter the da");
          double da = sc.nextDouble();
          System.out.println("Enter the years of experience");
          int years_of_exp = sc.nextInt();
          System.out.println("Enter the no of loans");
          int no_of_loans = sc.nextInt();
          double[] loan_amt = new double[no_of_loans];
          for(int j=0;j<no_of_loans;j++){
            System.out.println("Enter the loan amount "+(j+1));
            loan_amt[j] = sc.nextDouble();
          }
          sc.nextLine();
          p[i] = new
    personal(name,age,education,basic,hra,da,years_of_exp,no_of_loans,loan_amt);
          System.out.println("The details of the employee are: ");
          System.out.println("--------------------------------------");
          p[i].display(p[i]);
          System.out.println("--------------------------------------");
        }
      }
    }
```

Output

```
Enter the no of employees whose detail you wanna enter
2
Enter the name
John
Enter the age
29
Enter the education
BBA
Enter the basic salary200000
Enter the hra
2
Enter the da
3
Enter the years of experience
6
Enter the no of loans
2
Enter the loan amount 1
50000
Enter the loan amount 2
150000

The details of the employee are:
--------------------------------------
Name of the person is: John
Age of the person is: 29Education of the person is: BBA
The total salary of the person is: 200005.0
The years of experience of the person is: 6
The no of loans in the name of the person is: 2
The loan amount for each loan is:	50000.0 150000.0
The person is not eleigible for loan
No tax is to be paid
The person is not eleigible for promotion
--------------------------------------

Enter the name
Ram
Enter the age
45
Enter the education
```

```
MBA
Enter the basic salary
550000
Enter the hra
2
Enter the da
4
Enter the years of experience
12
Enter the no of loans
1
Enter the loan amount 1
250000

The details of the employee are:
--------------------------------------
Name of the person is: Ram
Age of the person is: 45
Education of the person is: MBA
The total salary of the person is: 550006.0
The years of experience of the person is: 12
The no of loans in the name of the person is: 1
The loan amount for each loan is:       250000.0
The person is eligible for loan
The tax to be paid is: 165001.8
The person is eligible for promotion
--------------------------------------
```

Q8. Create a Circle class with following members.
A data member that stores the radius of a circle.
A constructor function with an argument that initializes the radius
A function that computes and returns area of a circle
Create two derived classes Sector and Segment that inherit the Circle class. Both classes inherit radius and the function that returns the circle’s area from Circle. In addition to the members inherited from Circle, Sector and Segment have some specific members as follows:
Sector
1. A data member that stores the control angle of a sector (in radians)
2. A constructor function with arguments that initialize radius and angle
3. A function that computes and returns the area of a sector
Segment
1. A data member that stores the length of a segment in a circle
2. A constructor function with arguments that initialize radius and length
3. A function that computes and returns the area of a segment
Create the main () function to instantiate an object of each class and then call appropriate member functions to compute and return the area of a circle, sector and segment.

Note : Area_of_circle = pi * r^{2}

Area_of_Sector=(1/2) r^{2} *θ

Area_of_segment= r^{2} *((r-h)/r) – (r-h) $(2rh-h^{2})^{1/2}$ Where r is the radius of a circle, θ is the central angle of a sector in radians, h is the length of a segment and ((r-h)/r) is in radians.

```
import java.util.Scanner;
import java.lang.Math;
class Circle{
    double radius;
    Circle(double r){
        this.radius = r;
    }
    double circle_area(){
        return Math.PI*radius*radius;
    }
}
class Sector extends Circle{
    double angle; //To be taken in radians
    Sector(double r,double a){
        super(r);
        this.angle = a;
    }
    double sector_area(){
        return (0.5*radius*radius*angle);
    }
}
class Segment extends Circle{
    double length;
    Segment(double r,double l){
        super(r);
        this.length = l;
    }
    double segment_area(){
        return (((radius*radius)*Math.acos((radius-length)/radius)) -
((radius-length)*Math.sqrt(((2*radius*length)-(length*length)))));
    }
}
class inheritance{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("1. Area of circle");
        System.out.println("2. Area of sector");
        System.out.println("3. Area of segment");
        System.out.println("4. Exit!");
        while(true){
            System.out.println("Enter your choice");
            int ch = sc.nextInt();
            switch(ch){
                case 1:
                        System.out.println("Enter the radius of the circle");
                        double r = sc.nextDouble();
                        Circle c = new Circle(r);
                        System.out.println("The area of the circle with radius "+r+ " is
"+c.circle_area());
                        break;
                case 2:
                        System.out.println("Enter the radius of the circle");
                        r = sc.nextDouble();
                        System.out.println("Enter the sector angle in degrees");
                        double d = sc.nextDouble();
                        double p = (Math.PI/180)*d;
                        Sector s = new Sector(r,p);
                        System.out.println("The area of the sector with radius "+r+ " and sector
angle "+ d+"(degrees) is "+s.sector_area());
                        break;
                case 3:
                        System.out.println("Enter the radius of the circle");
                        r = sc.nextDouble();
                        System.out.println("Enter the length of the segment");
                        double l = sc.nextDouble();
                        Segment se = new Segment(r,l);
                        System.out.println("The area of the segment with radius "+r+ " and
segement length "+ l+" is "+se.segment_area());
                        break;
                case 4:
                        System.exit(0);
                        break;
                default:
                        System.out.println("Invalid choice");
            }
        }
    }
}
```

Output

```
1. Area of circle
2. Area of sector
3. Area of segment
4. Exit!

Enter your choice
1
Enter the radius of the circle
12
The area of the circle with radius 12.0 is 452.3893421169302

Enter your choice
2
Enter the radius of the circle
3
Enter the sector angle in degrees
45
The area of the sector with radius 3.0 and sector angle 45.0(degrees) is
3.5342917352885173

Enter your choice
3
Enter the radius of the circle
4
Enter the length of the segment
5
The area of the segment with radius 4.0 and segement length 5.0 is
33.048608657199026

Enter your choice
4
```

9.Write a Java Program that does the following related to Inheritance:

a. **Create an abstract class called Vehicle which contains the ‘year_of_manufacture’ data member and two abstract methods ‘getData()’ and ‘putData()’ with a constructor.**
b. **Create two derived classes “TwoWheeler” and “FourWheeler” and implement the abstract methods. Make “FourWheeler” as final class.**
c. **Create class ‘MyTwoWheeler’ which is a sub-class of “TwoWheeler” and demonstrate the use of super keyword to initialize data members of “MyTwoWheeler”.**

```
package program9 ;
public abstract class Vehicle {
  int  year_of_manufacture ;
  Vehicle()
  {
    year_of_manufacture = 0;
  }
  abstract int getData();
  abstract void putData(int year);
}
```

```
package program9 ;
class TwoWheeler extends Vehicle {
  int getData() {
    return year_of_manufacture;
  }
  void putData(int year) {
      year_of_manufacture = year;
  }
  TwoWheeler(int year) {
      year_of_manufacture = year;
  }
}
```

```
package program9 ;
final class FourWheeler extends Vehicle {
  int getData() {
      return year_of_manufacture;
  }
```

```
    void putData(int year) {
        year_of_manufacture = year;
    }
    FourWheeler(int year){
        year_of_manufacture=year;

    }
}

package program9 ;
class MyTwoWheeler extends TwoWheeler {
    MyTwoWheeler(int year) {
        super(year);
    }
}

package pragram9 ;
import java.util.*;
public class VehicleDriver {
    public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the year : ");
    int y = sc.nextInt();
    MyTwoWheeler myTwoWheeler=new MyTwoWheeler(2000);
        System.out.println(myTwoWheeler.getData());
    }
}
```

Output

```
Enter the year :
2000
2000
```

10. **Define an interface ‘Department’ with methods to readdata() and printdata(), print_number_designations(), number_research_consultancy_projs(). Define a ‘Faculty’ class with members name, designation, age, years of experience, joining_date and subjects_handled.**
    a. **In package ISE define the ‘ISE_department’ class that implements the ‘Department’ interface, accepts n faculty details and define all the methods. Raise a user defined exception ‘AgeException’ if the age of the faculty is > 58.**
    b. **In the default package define a ‘MainClass’ which uses the methods of the above classes and also displays those faculty details whose years of experience is greater than or equal to 20.**

```
package Q10;
import Q10.Faculty;
public interface department{

    void readData(Faculty f);
    void printData(Faculty f);
    void print_number_designations(Faculty f);
    void number_research_consultancy_projects(Faculty f);

}

package Q10;
public class Faculty{
   public String name,designation,joining_date,subjects_handled;
   public int age,years_of_experience,no_of_designations,no_of_research_projects;
}

package Q10;
import java.util.*;
import Q10.ISE.ISE_department;

public class MainClass{

    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int counter =0;
        int flag1 =0;
        int flag2 =0;
        int flag3 =0;
        int flag4 =0;
        ISE_department ise = new ISE_department();
        Faculty[] f = new Faculty[20];
        while(true){
            System.out.println("1. Add a faculty");
            System.out.println("2. Get the details of a particular faculty");
            System.out.println("3. Get the name of experienced faculty");
            System.out.println("4. Know the no of designations a particular faculty had");
            System.out.println("5. Know the no fo research consultancy proects done by a
faculty");
            System.out.println("6. Exit!");
            System.out.println("Enter your choice:\t");
            int ch = sc.nextInt();
            switch(ch){
                case 1:
                        f[counter] = new Faculty();
                        ise.readData(f[counter]);
                        counter++;
                        break;
                case 2:
                        System.out.println("Enter the name of the faculty whose details you
want");
                        sc.nextLine();
                        String x = sc.nextLine();
                        for(int i=0;i<counter;i++){
                            if(x.equals(f[i].name)){
                                flag1 = 1;
                                ise.printData(f[i]);
                                break;
                            }
                        }
                        if(flag1 == 0){
                            System.out.println("Faculty not found!");
                        }
                        break;
                case 3:
                        System.out.println("The faculty with experience greater than 20 years
are:");
                        for(int i=0;i<counter;i++){
                            if(f[i].years_of_experience >=20){
```

```
                System.out.println(f[i].name);
                flag2 = 1;
              }
            }
            if(flag2 == 0){
              System.out.println("No Such record found");
            }
            break;
        case 4:
            System.out.println("Enter the name of the faculty whose detail you want");
            sc.nextLine();
            x = sc.nextLine();
            for(int i=0;i<counter;i++){
              if(x.equals(f[i].name)){
                flag3 = 1;
                System.out.println("The no of designations the faculty had are:
"+f[i].no_of_designations);
                break;
              }
            }
            if(flag3 == 0){
              System.out.println("Faculty not found!");
            }
            break;
        case 5:
            System.out.println("Enter the name of the faculty whose detail you want");
            sc.nextLine();
            x = sc.nextLine();
            for(int i=0;i<counter;i++){
              if(x.equals(f[i].name)){
                flag4 = 1;
                System.out.println("The no of reserach consultancy projects faculty
did are: "+f[i].no_of_research_projects);
                break;
              }
            }
            if(flag4 == 0){
              System.out.println("Faculty not found!");
            }
            break;
        case 6:
            System.exit(0);
            break;
        default:
```

```
                        System.out.println("Invalid choice!");
            }
        }
    }
}

package Q10.ISE;
import java.util.Scanner;
import Q10.department;
import Q10.Faculty;
class MyException extends Exception{
    MyException(String s){
        super(s);
    }
}

package Q10.ISE;
import java.util.*;
import Q10.Faculty;
import Q10.department;
public class ISE_department implements department{

    public void readData(Faculty f){
        Scanner sc = new Scanner(System.in);
         try{
                System.out.println("Enter the name of the faculty");
                f.name = sc.nextLine();
                System.out.println("Enter the designation of the faculty");
                f.designation = sc.nextLine();
                System.out.println("Enter the Joining date of the faculty");
                f.joining_date = sc.nextLine();
                System.out.println("Enter the subjects handled by the faculty");
                f.subjects_handled = sc.nextLine();
                System.out.println("Enter the years of experience of the faculty");
                f.years_of_experience = sc.nextInt();
                System.out.println("Enter the no of designations the faculty had till date");
                f.no_of_designations = sc.nextInt();
                System.out.println("Enter the no of research consultancy projects done by the
faculty");
                f.no_of_research_projects = sc.nextInt();
                System.out.println("Enter the age of the faculty");
                int tempAge = sc.nextInt();
                if(tempAge<=58){
                    f.age = tempAge;
                }
                else{
                    throw new MyException("------------------AgeException: Should be less than
58 years-------------");
                }
            }
        catch(MyException ex){

                int age1;
                do{
                    System.out.println(ex.getMessage());
                    System.out.println("Enter the age of the faculty again:");
                    age1 = sc.nextInt();
                }while(age1>58);
                f.age = age1;
        }
    }

    public void printData(Faculty f){
        System.out.println("The name of the faculty is "+f.name);
        System.out.println("The age of the faculty is "+f.age);
        System.out.println("The designation of the faculty is "+f.designation);
        System.out.println("The joining date of the faculty is "+f.joining_date);
        System.out.println("The subjects handled by the faculty are: "+f.subjects_handled);
        System.out.println("The years of experience the faculty has is
"+f.years_of_experience);
        System.out.println("The number of designations faculty had are:
"+f.no_of_designations);
        System.out.println("The number of research consultancy projects are:
"+f.no_of_research_projects);
    }

    public void print_number_designations(Faculty f){
        System.out.println("The no of designations the faculty had are:
"+f.no_of_designations);
    }

    public void number_research_consultancy_projects(Faculty f){
        System.out.println("The number of research consultancy projects the faculty did
are: "+f.no_of_research_projects);
    }
}
```

Output

```
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
1
Enter the name of the faculty
Mr. John Victor
Enter the designation of the faculty
Associate Professor
Enter the Joining date of the faculty
02.10.2013
Enter the subjects handled by the faculty
Linear Algebra , Mathmatics
Enter the years of experience of the faculty
12
Enter the no of designations the faculty had till date
2
Enter the no of research consultancy projects done by the faculty
5
Enter the age of the faculty
43
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
1
Enter the name of the faculty
Ms. Arpita Mishra
Enter the designation of the faculty
Assistance Professor
Enter the Joining date of the faculty
05.08.2019
Enter the subjects handled by the faculty
Machine Learning , Data Science
Enter the years of experience of the faculty
```

```
5
Enter the no of designations the faculty had till date
1
Enter the no of research consultancy projects done by the faculty
6
Enter the age of the faculty
34
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
2
Enter the name of the faculty whose details you want
Ms. Arpita Mishra
The name of the faculty is Ms. Arpita Mishra
The age of the faculty is 34
The designation of the faculty is Assistance Professor
The joining date of the faculty is 05.08.2019
The subjects handled by the faculty are: Machine Learning , Data Science
The years of experience the faculty has is 5
The number of designations faculty had are: 1
The number of research consultancy projects are: 6
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
3
The faculty with experience greater than 20 years are:
No Such record found
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
4
Enter the name of the faculty whose detail you want
```

```
Mr. John Victor
The no of designations the faculty had are: 2
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
5
Enter the name of the faculty whose detail you want
Ms. Arpita Mishra
The no of reserach consultancy projects faculty did are: 6
1. Add a faculty
2. Get the details of a particular faculty
3. Get the name of experienced faculty
4. Know the no of designations a particular faculty had
5. Know the no fo research consultancy proects done by a faculty
6. Exit!
Enter your choice:
6
```

Q11.Write a Java Program that does the following related to Packages and Interfaces , Exception Handling:

a. **Create an interface Student which gets the name and branch of a student.**
b. **Create a package called 'StudentPackage' which has a user-defined class RegisterStudent.**
c. **If a student registers above 30 credits for the semester, the method should throw a user-defined exception called '*CreditLimit*' and display an appropriate message.**
d. **Create another package called 'ResultPackage' which displays the grade for the subject registered for particular semester and the SGPA . if SGPA is above 10 then throws an InvalidSGPA user-defined exception.**
e. **In the StudentPackage , collect the marks of all the subjects in 4 semesters and calculate SGPA and CGPA.**

```
package Q11.ResultPackage;
import java.util.Scanner;
import Q11.StudentPackage.CGPA;

class InvalidSGPA extends Exception{
   InvalidSGPA(String s){
      super(s);
   }
}
public class Result extends CGPA{

   public char grade[][] = new char[4][6];
   public int credit[] = new int[6];
   public String name ="";
   public String branch = "";
   public int semester;
   public String subjects[] = new String[6];
   public int totalCredits =0;
   Scanner sc = new Scanner(System.in);
   public void result(){
      for(int i=0;i<6;i++){
         System.out.println("Enter the marks in Subject["+(i+1)+"]");
```

```
    marks[semester-1][i] = sc.nextInt();
}

for(int i=0;i<6;i++){
    if(marks[semester-1][i]>90){
        grade[semester-1][i] = 'O';
        sgpa[semester-1] += credit[i]*10;
    }

    else if(marks[semester-1][i]>80){
        grade[semester-1][i] = 'S';
        sgpa[semester-1] += credit[i]*9;
    }

    else if(marks[semester-1][i]>70){
        grade[semester-1][i] = 'A';
        sgpa[semester-1] += credit[i]*8;
    }

    else if(marks[semester-1][i]>60){
        grade[semester-1][i] = 'B';
        sgpa[semester-1] += credit[i]*7;
    }

    else if(marks[semester-1][i]>50){
        grade[semester-1][i] = 'C';
        sgpa[semester-1] += credit[i]*6;
    }

    else if(marks[semester-1][i]>40){
        grade[semester-1][i] = 'D';
        sgpa[semester-1] += credit[i]*5;
    }
    else{
        grade[semester-1][i] = 'F';
        sgpa[semester-1] += credit[i]*5;
    }
}
sgpa[semester-1] /= totalCredits;
```

```
        try{
            if(sgpa[semester-1]>10){
                throw new InvalidSGPA("Invalid SGPA Exception");
            }
            else{
                System.out.println("SGPA in semester"+semester+" = "+sgpa[semester-1]);
            }
        }
        catch(InvalidSGPA ex){
            System.out.println(ex.getMessage());
        }
    }
}
```

```
package Q11.StudentPackage;
public class CGPA{
    public int marks[][] = new int[4][6];
    public float sgpa[] = new float[4];
    float cgpa =0;
    public void calculateGpa(){
        for(int i=0;i<4;i++){
            cgpa += sgpa[i];
        }
        cgpa /=4;
        System.out.println("CGPA is:"+cgpa);
    }
}
```

```
package Q11.StudentPackage;
import java.util.Scanner;
import Q11.ResultPackage.Result;

interface Student{
    public void getName();
    public void getBranch();
}
class CreditLimit extends Exception{
    CreditLimit(String s){
        super(s);
    }
}
public class RegisterStudent extends Result implements Student{
    Scanner sc = new Scanner(System.in);
    public void getName(){
        System.out.println("Enter name:\t");
        name = sc.nextLine();
    }
    public void getBranch(){
        System.out.println("Enter branch:\t");
        branch = sc.nextLine();
    }
    public void register(int sem){
        semester = sem;
        if(semester == 1){
                getName();
                getBranch();
        }
        System.out.println("Enter the 6 subjectss registered");
        for(int i=0;i<6;i++){
            System.out.println("Enter subjects "+(i+1));
            subjects[i] = sc.next();
        }
        System.out.println("Enter the respective credits");
        totalCredits =0;
        for(int i=0;i<6;i++){
            System.out.println("Enter the credits of subjects "+(i+1));
            credit[i] = sc.nextInt();
            totalCredits += credit[i];
        }
        try{
            if(totalCredits >30){
                throw new CreditLimit("Credit Limit Exception");
            }
        }
        catch(CreditLimit cl){
            System.out.println(cl.getMessage());
        }
    }
```

```
}
```

```
package Q11;
import Q11.StudentPackage.RegisterStudent;
import Q11.StudentPackage.CGPA;
class MainClass{
    public static void main(String[] args){
        RegisterStudent s = new RegisterStudent();
        System.out.println("For Semester 1");
        s.register(1);
        s.result();
        System.out.println("For Semester 2");
        s.register(2);
        s.result();
        System.out.println("For Semester 3");
        s.register(3);
        s.result();
        System.out.println("For Semester 4");
        s.register(4);
        s.result();
        s.calculateGpa();

    }
}
```

Output

```
For Semester 1
Enter name:
Ram Mishra
Enter branch:
ISE
Enter the 6 subjects registered
Enter subjects 1
Civil
Enter subjects 2
Maths
Enter subjects 3
Mechanical
Enter subjects 4
Data Science
Enter subjects 5
DIP
Enter subjects 6
AI
Enter the respective credits
Enter the credits of subjects 1
3
Enter the credits of subjects 2
3
Enter the credits of subjects 3
4
Enter the credits of subjects 4
2
Enter the credits of subjects 5
1
Enter the credits of subjects 6
3
Enter the marks in Subject[1]
78
Enter the marks in Subject[2]
89
Enter the marks in Subject[3]
76
Enter the marks in Subject[4]
```

```
86
Enter the marks in Subject[5]
90
Enter the marks in Subject[6]
67
SGPA in semester1 = 8.1875
For Semester 2
Enter the 6 subjects registered
Enter subjects 1
CN
Enter subjects 2
Microcontrollers
Enter subjects 3
Blockchain
Enter subjects 4
Cloud Computing
Enter subjects 5
OOPS
Enter subjects 6
ML
Enter the respective credits
Enter the credits of subjects 1
3
Enter the credits of subjects 2
4
Enter the credits of subjects 3
3
Enter the credits of subjects 4
1
Enter the credits of subjects 5
4
Enter the credits of subjects 6
3
Enter the marks in Subject[1]
76
Enter the marks in Subject[2]
45
Enter the marks in Subject[3]
56
Enter the marks in Subject[4]
76
Enter the marks in Subject[5]
65
Enter the marks in Subject[6]
```

```
89
SGPA in semester2 = 6.9444447
For Semester 3
Enter the 6 subjects registered
Enter subjects 1
Java
Enter subjects 2
C programing
Enter subjects 3
SE
Enter subjects 4
IPR
Enter subjects 5
Hospital Manangement
Enter subjects 6
IOT
Enter the respective credits
Enter the credits of subjects 1
2
Enter the credits of subjects 2
3
Enter the credits of subjects 3
4
Enter the credits of subjects 4
2
Enter the credits of subjects 5
1
Enter the credits of subjects 6
4
Enter the marks in Subject[1]
89
Enter the marks in Subject[2]
76
Enter the marks in Subject[3]
78
Enter the marks in Subject[4]
76
Enter the marks in Subject[5]
70
Enter the marks in Subject[6]
90
SGPA in semester3 = 8.3125
For Semester 4
Enter the 6 subjectss registered
Enter subjects 1
Math-II
Enter subjects 2
Kannada
Enter subjects 3
OR
Enter subjects 4
DBMS
Enter subjects 5
Discrete Mathematics
Enter subjects 6
Enter the respective credits
Enter the credits of subjects 1
3
Enter the credits of subjects 2
3
Enter the credits of subjects 3
2
Enter the credits of subjects 4
3
Enter the credits of subjects 5
4
Enter the credits of subjects 6
3
Enter the marks in Subject[1]
86
Enter the marks in Subject[2]
67
Enter the marks in Subject[3]
75
Enter the marks in Subject[4]
90
Enter the marks in Subject[5]
99
Enter the marks in Subject[6]
89
SGPA in semester4 = 8.777778
CGPA is:8.055555
```

Q12. a. Write a java program to implement queues of Strings using an ArrayList class of the Collection framework.

```
import java.util.*;
class ArrayList1{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        ArrayList<String> Q = new ArrayList<String>();
        while(true){
            System.out.println("1. Enqueue");
            System.out.println("2. Dequeue");
            System.out.println("3. Peak Element");
            System.out.println("4. Display");
            System.out.println("5. Exit!");
            System.out.println("Enter your choice");
            int ch = sc.nextInt();
            switch(ch){
                case 1:
                        System.out.println("Enter the string you wanna Enqueue");
                        String x = sc.next();
                        Q.add(Q.size(),x);
                        System.out.println("Element enqueued successfully");
                        break;
                case 2:
                        if(Q.isEmpty()){
                            System.out.println("The Queue is empty");
                        }
                        else{
                            System.out.println("The element dequeued is: "+Q.remove(0));
                        }
                        break;
                case 3:
                        if(Q.isEmpty()){
                            System.out.println("The Queue is empty");
                        }
                        else{
                            System.out.println("The element at the peak is :"+Q.get(0));
                        }
                        break;
                case 4:
                        if(Q.isEmpty()){
                            System.out.println("The Queue is empty");
                        }
                        else{
                            System.out.println("The elements of the queue are:");
                            for(int i=0;i<Q.size();i++){
                                System.out.println(Q.get(i));
                            }
                        }
                        break;
                case 5:
                        System.exit(0);
                        break;
                default:
                        System.out.println("Invalid choice!");
            }
    }
}
}
```

Output

```
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
1
Enter the string you wanna Enqueue
Computer
Element enqueued successfully
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
1
Enter the string you wanna Enqueue
Science
Element enqueued successfully
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
1
Enter the string you wanna Enqueue
Department
Element enqueued successfully
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
```

```
4
The elements of the queue are:
Computer
Science
Department
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
2
The element dequeued is: Computer
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice3
The element at the peak is : Science
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
4
The elements of the queue are:
Science
Department
1. Enqueue
2. Dequeue
3. Peak Element
4. Display
5. Exit!
Enter your choice
5
```

Q12.b.Create a linked list of names (String type). Use an Iterator to traverse through the list and print those names whose length is < 5.

```
import java.util.Scanner;
import java.util.LinkedList;
import java.util.Iterator;
class LinkedList1{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        LinkedList<String> ll = new LinkedList<String>();
        while(true){
            System.out.println("1. Insert an element into the linked list");
            System.out.println("2. Remove an element from the linked list");
            System.out.println("3. List all elements with length less than 5");
            System.out.println("4. Display the LinkedList");
            System.out.println("5. Exit!");
            System.out.print("Enter your choice:\t");
            int ch = sc.nextInt();
            switch(ch){
                case 1:
                        System.out.println("Enter the element to be inserted");
                        String x = sc.next();
                        ll.add(x);
                        System.out.println("Element inserted successfully");
                        break;
                case 2:
                        if(ll.peek() == null){
                            System.out.println("The linked list is empty");
                        }
                        else{
                            System.out.println("Enter the index of the element you wanna
remove");
                            int p = sc.nextInt();
                            System.out.println("The elemnet removed is: "+ll.remove(p));
                        }
                        break;
                case 3:
                        if(ll.peek() == null){
                            System.out.println("The Linked list is empty");
            }
            else{
              Iterator<String> itr = ll.iterator();
              String e1 = "";
              System.out.println("The strings with length less than 5 are: ");
              while(itr.hasNext()){
                e1 = itr.next();
                if(e1.length()<5){
                  System.out.println(e1);
                }
              }
            }
            break;
        case 4:
            if(ll.peek() == null){
              System.out.println("The linked list is empty");
            }
            else{
              Iterator<String> itr = ll.iterator();
              System.out.println("The elements of the linked list are:");
              while(itr.hasNext()){
                System.out.println(itr.next());
              }
            }
            break;
        case 5:
            System.exit(0);
            break;
        default:
            System.out.println("Invalid choice");
      }
    }
  }
}
```

Output

```
1. Insert an element into the linked list
2. Remove an element from the linked list
3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      1
Enter the element to be inserted
item1
Element inserted successfully
1. Insert an element into the linked list
2. Remove an element from the linked list
3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      1
Enter the element to be inserted
item2
Element inserted successfully
1. Insert an element into the linked list
2. Remove an element from the linked list
3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      1
Enter the element to be inserted
item2
Element inserted successfully
1. Insert an element into the linked list
2. Remove an element from the linked list
3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      2
Enter the index of the element you wanna remove
2
```

```
The element removed is: item2
1. Insert an element into the linked list
2. Remove an element from the linked list
3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      3
The strings with length less than 5 are:

1. Insert an element into the linked list
2. Remove an element from the linked list
3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      4
The elements of the linked list are:
item1
Item2
1. Insert an element into the linked list
2. Remove an element from the linked list3. List all elements with length less than 5
4. Display the LinkedList
5. Exit!
Enter your choice:      5
```