Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
package ds2021;
import java.util.Scanner;
public class SinglyLinkedList{
    class Node{
        int roll_no;
        String name, mail_id;
        float per;
        Node next;
        public Node(int roll_no, String name, String mail_id, float per ){
            this.roll_no = roll_no;
            this.name = name;
            this.mail_id = mail_id;
            this.per = per;
            this.next = null;
        }
    }
    // head pointer
    Node head=null;

    //IsEmpty
    boolean isEmpty(){
        if(head==null)
            return true;
        else
            return false;
    }

    //Insert at end
    void InsertAtEnd(int roll_no, String name, String mail_id, float per){
        Node nn = new Node(roll_no, name, mail_id, per);
        if(isEmpty()){
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
            head = nn;
            System.out.println("Empty Linked List So Roll no :  "+roll_no+" is Inserted At Front of Linked List");
        }
        else{
            Node temp = head;
            while(temp.next != null){
                temp  = temp.next;
            }
            temp.next = nn;
            System.out.println("Roll no :  "+roll_no+" is Inserted At End of Linked List");
        }
    }

    //Insert before roll_no
    void InsertBeforeVal(int old_roll_no, int roll_no, String name, String mail_id, float per){
        Node nn = new Node(roll_no, name, mail_id, per);
        if(isEmpty()){
            System.out.println("Linked List is Empty..( "+roll_no+" is Inserted At Head.)");
            head = nn;
        }
        else if(old_roll_no == head.roll_no){
            nn.next = head;
            head = nn;
            System.out.println(" Old Roll no :  "+old_roll_no+" is located At HEAD (1st) Location..( "+roll_no+" is
Inserted At Head.)");
        }
        else{
            Node temp = head;
            while(temp.next != null){
                if(old_roll_no == temp.next.roll_no){
                    break;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
            }
            temp  = temp.next;
        }
        if(temp.next!=null){
            nn.next = temp.next;
            temp.next = nn;
                    System.out.println("Old Roll no :  "+roll_no+" is Inserted before "+old_roll_no+" Students
List..");
        }
        else{
            System.out.println("Old Roll no :  "+old_roll_no+" is Not in Students List..(Inserted At End of Linked
List)");
            temp.next = nn;
        }
    }
  }

  //Insert after roll_no
  void InsertAfterVal(int old_roll_no, int roll_no, String name, String mail_id, float per){
    Node nn = new Node(roll_no, name, mail_id, per);
    if(isEmpty()){
        System.out.println("Linked List is Empty..( "+roll_no+" is Inserted At Head.)");
        head = nn;
    }
    else{
        Node temp = head;
        while(temp.next != null){
            if(old_roll_no == temp.roll_no){
                break;
            }
            temp  = temp.next;
        }
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
        if(temp.next != null){
            nn.next = temp.next;
            temp.next = nn;
        }
        else{
            System.out.println("Postion Not....(Inserted At End of Linked List)");
            temp.next = nn;
        }
    }
}

//Insert at pos
void InsertAtPos(int pos, int roll_no, String name, String mail_id, float per){
    Node nn = new Node(roll_no, name, mail_id, per);
    if(pos==1){
        //this.InsertAtBegin(val);
            nn.next = head;
            head = nn;
    }
    else if(pos==0){
       System.out.println("Position should be start from 1...");
    }
    else{
        if(isEmpty()){
            System.out.println("Linked List is Empty..( Inserted At Head.)");
            head = nn;
            return;
        }
        int count = 1;
        Node temp = head;
        while(temp.next != null && count<pos-1){
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
            count++;
            temp  = temp.next;
        }
        if(temp.next != null){
            nn.next = temp.next;
            temp.next = nn;
        }
        else{
            System.out.println("Postion Not....( "+roll_no+" is Inserted At End.)");
            temp.next = nn;
        }
    }
}

//Delete by name
void DeleteByName(String name){
    if(isEmpty()){
       System.out.println("Linked List is Empty..");
    }
    else if(name.compareToIgnoreCase(del_name) == 0)
    {
        head = head.next;
              System.out.println("Name : " +name+"is deleted.");
        return;
    }
    else{
        Node temp = head;
              String del_name = " ";
        while(temp.next != null){
            if(name.compareToIgnoreCase(del_name) == 0){
                del_name = temp.next.name;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
                temp.next = temp.next.next;
                break;
            }
            temp  = temp.next;
        }
        if(temp.next !=null || name.compareToIgnoreCase(del_name) == 0){
            System.out.println("Name : " +name+ " is deleted.");
        }
        else{
            System.out.println("Name "+name+" is Not Located in Student List..");
        }
    }
}

//Display
void Display(){
    Node temp=head;
    System.out.println("Students are : ");
    System.out.println("roll_no \t:\t name \t:\t mail_id \t:\t percentages" );
    while(temp!=null){
        System.out.println(temp.roll_no+"\t:\t"+temp.name+"\t:\t"+temp.mail_id+"\t:\t"+temp.per);
        temp=temp.next;
    }
}
public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    SinglyLinkedList l = new SinglyLinkedList();
    int pos, old_roll_no,roll_no;
    String name, mail_id;
    float per;
    char c,ch;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
String s;
while(true){
    System.out.println("=================");
    System.out.println("Options");
    System.out.println("=================");
    System.out.println("1 - Insert at last");
    System.out.println("2 - Insert at given position");
    System.out.println("3 - Delete by name");
    System.out.println("4/<p> - Display");
    System.out.println("0/<q> - Exit");
    System.out.println("Enter Your Choice : ");
    s = scn.next();
    c = s.charAt(0);
    switch(c){
        case '1':
            System.out.println("Enter Roll_no : ");
            roll_no = scn.nextInt();
            System.out.println("Enter Name : ");
            name=scn.next();
            System.out.println("Enter Mail_id : ");
            mail_id=scn.next();
            System.out.println("Enter Percentage : ");
            per=scn.nextFloat();
            l.InsertAtEnd(roll_no, name, mail_id, per);
            break;
        case '2':
            System.out.println("=================");
            System.out.println("Options for Insertion at position : ");
            System.out.println("1 - Before Roll_no");
            System.out.println("2 - After Roll_no");
            System.out.println("3 - At position");
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
System.out.println("Enter  Your Choice : ");
s = scn.next();
ch = s.charAt(0);
switch(ch){
   case '1':
      System.out.println("Enter Roll_no (before which you have to insert) : ");
      old_roll_no=scn.nextInt();
      System.out.println("Enter Roll_no : ");
      roll_no=scn.nextInt();
      System.out.println("Enter Name : ");
      name=scn.next();
      System.out.println("Enter Mail_id : ");
      mail_id=scn.next();
      System.out.println("Enter Percentage : ");
      per=scn.nextFloat();
      l.InsertBeforeVal(old_roll_no, roll_no, name, mail_id, per);
      break;
   case '2':
      System.out.println("Enter Roll_no (after which you have to insert) : ");
      old_roll_no=scn.nextInt();
      System.out.println("Enter Roll_no : ");
      roll_no=scn.nextInt();
      System.out.println("Enter Name : ");
      name=scn.next();
      System.out.println("Enter Mail_id : ");
      mail_id=scn.next();
      System.out.println("Enter Percentage : ");
      per=scn.nextFloat();
      l.InsertAfterVal(old_roll_no, roll_no, name, mail_id, per);
      break;
   case '3':
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
                System.out.println("Enter Position (which position you have to insert) : ");
                pos=scn.nextInt();
                System.out.println("Enter Roll_no : ");
                roll_no=scn.nextInt();
                System.out.println("Enter Name : ");
                name=scn.next();
                System.out.println("Enter Mail_id : ");
                mail_id=scn.next();
                System.out.println("Enter Percentage : ");
                per=scn.nextFloat();
                l.InsertAtPos(pos, roll_no, name, mail_id, per);
                break;
            default :
                System.out.println("Please select valid Choice....");
                break;
        }
        break;
    case '3':
      if(l.isEmpty()){
          System.out.println("Student List is Empty....");
          break;
        }
        System.out.println("Enter Name : ");
        name=scn.next();
        l.DeleteByName(name);
        break;
    case 'p':
    case '4':
        if(l.isEmpty()){
            System.out.println("Student List is Empty....");
            break;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q1. : Write a menu driven program in Java to perform following operations on a singly linked list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and percentage)
1. Insert (insert at last)
2. Insert at given position
3. Delete (delete record with given name)
4. Display (display information for all students)

```
                }
                l.Display();
                break;
            case 'q':
            case '0':
                System.exit(0);
                break;
            default:
                System.out.println("Please enter valid choice....");
                break;
        }
    }
  }
}
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program in Java to create a singly linked list in sorted (ascending) order of. Provide following operation on link list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and marks)
1. Insert a student record (at proper place to maintain tree in sorted order)
2. Delete a record for given roll no
3. Display all records

```
package ds2021;
import java.util.Scanner;
public class SortedLinkedList{
    class Node{
        int roll_no;
        String name, mail_id;
        int marks;
        Node next;
        public Node(int roll_no, String name, String mail_id, int marks ){
            this.roll_no = roll_no;
            this.name = name;
            this.mail_id = mail_id;
            this.marks = marks;
            this.next = null;
        }
    }
    // head pointer
    Node head=null;
    //IsEmpty
    boolean isEmpty(){
        if(head==null)
            return true;
        else
            return false;
    }
    //Insert before marks (Asecending Sorted Order)
    void InsertBeforeVal(int roll_no, String name, String mail_id, int marks){
        Node nn = new Node(roll_no, name, mail_id, marks);
        if(isEmpty()){
            System.out.println("Linked List is Empty..( "+roll_no+" is Inserted At Head.)");
            head = nn;
        }
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program in Java to create a singly linked list in sorted (ascending) order of. Provide following operation on link list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and marks)
1. Insert a student record (at proper place to maintain tree in sorted order)
2. Delete a record for given roll no
3. Display all records

```
    else if(marks < head.marks){
        nn.next = head;
        head = nn;
        System.out.println(" Roll no :  "+roll_no+" is Inserted At Head.)");
    }
    else{
        Node temp = head;
        while(temp.next != null){
            if(marks < temp.next.marks){
                break;
            }
            temp  = temp.next;
        }
        if(temp.next!=null){
            nn.next = temp.next;
            temp.next = nn;
            System.out.println("Roll no :  "+roll_no+" is inserted..");
        }
        else{
            System.out.println("Roll no :  "+roll_no+" is inserted at End of Linked List)");
            temp.next = nn;
        }
    }
}
//Delete by rollno
void DeleteByRollno(int old_roll_no){
    if(isEmpty()){
       System.out.println("Linked List is Empty..");
    }
    else if(head.roll_no == old_roll_no)
    {
        head = head.next;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program in Java to create a singly linked list in sorted (ascending) order of. Provide following operation on link list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and marks)
1. Insert a student record (at proper place to maintain tree in sorted order)
2. Delete a record for given roll no
3. Display all records

```
            System.out.println("Roll no  :" +old_roll_no+"is deleted.");
            return;
        }
        else{
            Node temp = head;
            int del_roll_no = 0;
            while(temp.next != null){
                if(temp.next.roll_no == old_roll_no){
                    del_roll_no = temp.next.roll_no;
                    temp.next = temp.next.next;
                    break;
                }
                temp  = temp.next;
            }
            if(temp.next !=null || del_roll_no == old_roll_no){
                System.out.println("Roll no  :" +old_roll_no+"is deleted.");
            }
            else{
                System.out.println("Roll no : "+old_roll_no+" is Not Located in Student List..");
            }
        }
    }
    //Display
    void Display(){
        Node temp=head;
        System.out.println("Students are : ");
        System.out.printf("\n roll_no \t name \t\t mail_id \t\t marks \n");
        while(temp!=null){
            System.out.println(temp.roll_no+"\t\t"+temp.name+"\t\t"+temp.mail_id+"\t\t"+temp.marks);
            temp=temp.next;
        }
    }
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program in Java to create a singly linked list in sorted (ascending) order of. Provide following operation on link list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and marks)
1. Insert a student record (at proper place to maintain tree in sorted order)
2. Delete a record for given roll no
3. Display all records

```
public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    SortedLinkedList l = new SortedLinkedList();
    int roll_no;
    String name, mail_id;
    int marks;
    char c,ch;
    String s;
    while(true){
        System.out.println("=================");
        System.out.println("Options");
        System.out.println("=================");
        System.out.println("1 - Insert");
        System.out.println("2 - Delete by roll_no");
        System.out.println("3/<p> - Display");
        System.out.println("0/<q> - Exit");
        System.out.println("Enter Your Choice : ");
        s = scn.next();
        c = s.charAt(0);
        switch(c){
            case '1':
                System.out.println("Enter Roll_no : ");
                roll_no = scn.nextInt();
                System.out.println("Enter Name : ");
                name=scn.next();
                System.out.println("Enter Mail_id : ");
                mail_id=scn.next();
                System.out.println("Enter marks : ");
                marks=scn.nextInt();
                l.InsertBeforeVal(roll_no, name, mail_id, marks);
                break;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q2. : Write a menu driven program in Java to create a singly linked list in sorted (ascending) order of. Provide following operation on link list. Each elements of link list contains information for individual student i.e. (roll no, name, mail_id and marks)
1. Insert a student record (at proper place to maintain tree in sorted order)
2. Delete a record for given roll no
3. Display all records

```
            case '2':
              if(l.isEmpty()){
                  System.out.println("Student List is Empty....");
                  break;
              }
              System.out.println("Enter Roll no : ");
              roll_no=scn.nextInt();
              l.DeleteByRollno(roll_no);
              break;
            case 'p':
            case '3':
              if(l.isEmpty()){
                  System.out.println("Student List is Empty....");
                  break;
              }
              l.Display();
              break;
            case 'q':
            case '0':
              System.exit(0);
              break;
            default:
              System.out.println("Please enter valid choice....");
              break;
        }
     }
   }
}
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
package ds2021;
import java.util.Scanner;
public class DoublyLinkedList{
    class Node{
        int book_id;
        String book_title;
        float price;
        Node prev,next;
        public Node(int book_id, String book_title, float price){
            this.book_id = book_id;
            this.book_title = book_title;
            this.price = price;
            this.prev = null;
            this.next = null;
        }
    }
    // head pointer
    Node head=null;

    //IsEmpty
    boolean isEmpty(){
        if(head==null)
            return true;
        else
            return false;
    }

     //Insert at begin
    void InsertAtBegin(int book_id, String book_title, float price){
        Node nn = new Node(book_id, book_title, price);
        if(isEmpty()){
            head = nn;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
            System.out.println("Empty Linked List So Book_Id :  "+book_id+" is Inserted At Begin of Linked List");
        }
        else{
            head.prev = nn;
            nn.next = head;
            head = nn;
            System.out.println("Book_Id :  "+book_id+" is Inserted At Begin of Linked List");
        }
    }

    //Insert at end
    void InsertAtEnd(int book_id, String book_title, float price){
        Node nn = new Node(book_id, book_title, price);
        if(isEmpty()){
            head = nn;
            System.out.println("Empty Linked List So Book_Id :  "+book_id+" is Inserted At Front of Linked List");
        }
        else{
            Node temp = head;
            while(temp.next != null){
                temp  = temp.next;
            }
            temp.next = nn;
            nn.prev = temp;
            System.out.println("Book_Id :  "+book_id+" is Inserted At End of Linked List");
        }
    }

    //Insert before book_id
    void InsertBeforeVal(int old_book_id, int book_id, String book_title, float price){
        Node nn = new Node(book_id, book_title, price);
        if(isEmpty()){
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
            System.out.println("Linked List is Empty..( "+book_id+" is Inserted At Head.)");
            head = nn;
        }
        else if(old_book_id == head.book_id){
            head.prev = nn;
            nn.next = head;
            head = nn;
            System.out.println(book_id+" is Inserted At Head.)");
        }
        else{
            Node temp = head;
            while(temp.next != null){
                if(old_book_id == temp.book_id){
                    break;
                }
                temp  = temp.next;
            }
            if(temp.next!=null || temp.book_id == old_book_id){
                nn.next = temp;
                nn.prev = temp.prev;
                temp.prev.next = nn;
                temp.prev = nn;
                        System.out.println("Old Book_Id :  "+book_id+" is Inserted before "+old_book_id+" Books
List..");
            }
            else{
                System.out.println("Inserted At End of Linked List)");
                temp.next = nn;
                nn.prev = temp;
            }
        }
    }
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
//Insert after book_id
void InsertAfterVal(int old_book_id, int book_id, String book_title, float price){
    Node nn = new Node(book_id, book_title, price);
    if(isEmpty()){
        System.out.println("Linked List is Empty..( "+book_id+" is Inserted At Head.)");
        head = nn;
    }
    else{
        Node temp = head;
        while(temp.next != null){
            if(old_book_id == temp.book_id){
                break;
            }
            temp  = temp.next;
        }
        if(temp.next != null || temp.book_id == old_book_id){
            nn.next = temp.next;
            nn.prev = temp;
            if(temp.next != null)
            {
                temp.next.prev = nn;
            }
            temp.next = nn;
        }
        else{
            System.out.println("Inserted At End of Linked List)");
            temp.next = nn;
            nn.prev = temp;
        }
    }
}
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
//Delete by book_id
void DeleteBybook_id(int book_id){
    if(isEmpty()){
       System.out.println("Linked List is Empty..");
    }
    else{
        Node temp = head;
              int del_book_id = 0;
        while(temp != null && temp.book_id != book_id){

            temp  = temp.next;
        }
        if(temp != null){
            del_book_id = temp.book_id;
            if(temp.prev != null){
                temp.prev.next = temp.next;
            }
            else{
                head = head.next;
                head.prev = null;
            }
            if(temp.next != null)
            {
                temp.next.prev = temp.prev;
            }
            else
            {
                temp.prev.next = null;
            }
            System.out.println("book_id : " +del_book_id+ " is deleted.");
        }
        else{
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
            System.out.println("book_id "+book_id+" is Not Located in Book List..");
        }
    }
}

//Search
void Search(String book_title){
    Node temp=head;
    System.out.println("Books are : ");
    System.out.println("book_id \t:\t book_title \t:\t price" );
    while(temp!=null){
        if(book_title.compareToIgnoreCase(temp.book_title) == 0)
            System.out.println(temp.book_id+"\t:\t"+temp.book_title+"\t:\t"+temp.price);
        temp=temp.next;
    }
}

//Display
void Display(){
    Node temp=head;
    System.out.println("Books are : ");
    System.out.println("book_id \t:\t book_title \t:\t price" );
    while(temp!=null){
        System.out.println(temp.book_id+"\t:\t"+temp.book_title+"\t:\t"+temp.price);
        temp=temp.next;
    }
}

public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    DoublyLinkedList l = new DoublyLinkedList();
    int pos, old_book_id,book_id;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
String book_title;
float price;
char c,ch;
String s;
while(true){
   System.out.println("================");
   System.out.println("Options");
   System.out.println("================");
   System.out.println("1 - Insert");
   System.out.println("2 - Delete by book_id");
   System.out.println("3 - Search");
   System.out.println("4/<p> - Display");
   System.out.println("0/<q> - Exit");
   System.out.println("Enter Your Choice : ");
   s = scn.next();
   c = s.charAt(0);
   switch(c){
      case '1':
         System.out.println("================");
         System.out.println("Options for Insertion  : ");
         System.out.println("1 - At Begin");
         System.out.println("2 - At End");
         System.out.println("3 - Before Book_id");
         System.out.println("4 - After Book_id");
         System.out.println("Enter  Your Choice : ");
         s = scn.next();
         ch = s.charAt(0);
         switch(ch){
            case '1':
               System.out.println("Enter book_id : ");
               book_id = scn.nextInt();
               System.out.println("Enter book_title : ");
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
            book_title=scn.next();
            System.out.println("Enter price : ");
            price=scn.nextFloat();
            l.InsertAtBegin(book_id, book_title, price);
            break;
        case '2':
            System.out.println("Enter book_id : ");
            book_id = scn.nextInt();
            System.out.println("Enter book_title : ");
            book_title=scn.next();
            System.out.println("Enter price : ");
            price=scn.nextFloat();
            l.InsertAtEnd(book_id, book_title, price);
            break;
        case '3':
            System.out.println("Enter Book_id (Before which you have to insert) : ");
            old_book_id=scn.nextInt();
            System.out.println("Enter book_id : ");
            book_id = scn.nextInt();
            System.out.println("Enter book_title : ");
            book_title=scn.next();
            System.out.println("Enter price : ");
            price=scn.nextFloat();
            l.InsertBeforeVal(old_book_id, book_id, book_title, price);
            break;
        case '4':
            System.out.println("Enter Book_id (After which you have to insert) : ");
            old_book_id=scn.nextInt();
            System.out.println("Enter book_id : ");
            book_id = scn.nextInt();
            System.out.println("Enter book_title : ");
            book_title=scn.next();
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
                System.out.println("Enter price : ");
                price=scn.nextFloat();
                l.InsertAfterVal(old_book_id, book_id, book_title, price);
                break;
            default :
                System.out.println("Please select valid Choice....");
                break;
        }
        break;
    case '2':
       if(l.isEmpty()){
            System.out.println("Book List is Empty....");
            break;
        }
        System.out.println("Enter book_id : ");
        book_id=scn.nextInt();
        l.DeleteBybook_id(book_id);
        break;
    case '3':
       if(l.isEmpty()){
            System.out.println("Book List is Empty....");
            break;
        }
        System.out.println("Enter book_title : ");
        book_title=scn.next();
        l.Search(book_title);
        break;
    case 'p':
    case '4':
        if(l.isEmpty()){
            System.out.println("Book List is Empty....");
            break;
```

Roll No. : 124
Name : Singh Chandani Harendra

Q3. : Write a menu driven program in Java to perform following operations on doubly linked list. Each elements of link list contains information for individual book in library i.e. (BookID, BookTitle, Price) .
1. Insert a record
2. Delete (delete record with given BookID)
3. Search (display all books with give title)
4 Display (display information for all students)

```
                }
                l.Display();
                break;
            case 'q':
            case '0':
                System.exit(0);
                break;
            default:
                System.out.println("Please enter valid choice....");
                break;
        }
    }
  }
}
```

Roll No. : 124
Name : Singh Chandani Harendra

Q4. : Write a menu driven program in Java to create a singly circular linked list of student information. Each elements of link list contains information for individual student i.e. (roll no, name marks) provides following operation.
(1) Insert
(2) Search and display (find a node with given name and display all records of list traversing the entire link circularly from searched node.)

```
package ds2021;
import java.util.Scanner;
public class SinglyCircularLinkedList{
    class Node{
        int roll_no;
        String name, mail_id;
        int marks;
        Node next;
        public Node(int roll_no, String name, String mail_id, int marks ){
            this.roll_no = roll_no;
            this.name = name;
            this.mail_id = mail_id;
            this.marks = marks;
            this.next = null;
        }
    }
    // head pointer
    Node head=null;
    //IsEmpty
    boolean isEmpty(){
        if(head==null)
            return true;
        else
            return false;
    }
    //Insert at end
    void InsertAtEnd(int roll_no, String name, String mail_id, int marks){
        Node nn = new Node(roll_no, name, mail_id, marks);
        if(isEmpty()){
            head = nn;
            nn.next = head;  // circular
            System.out.println("Empty Linked List So Roll no :  "+roll_no+" is Inserted At Front of Linked List");
```

Roll No. : 124
Name : Singh Chandani Harendra

Q4. : Write a menu driven program in Java to create a singly circular linked list of student information. Each elements of link list contains information for individual student i.e. (roll no, name marks) provides following operation.
(1) Insert
(2) Search and display (find a node with given name and display all records of list traversing the entire link circularly from searched node.)

```
    }
    else{
        Node temp = head;
        while(temp.next != head){
            temp  = temp.next;
        }
        nn.next = temp.next; //circular
        temp.next = nn;
        System.out.println("Roll no :  "+roll_no+" is Inserted At End of Linked List");
    }
}
//Search and Display
void Search_Display(String search_name){
    Node temp=head;
    Node current=head;
    if(search_name.compareToIgnoreCase(head.name) == 0){
        current = head ;
    }
    else{
        temp = temp.next;
        while(temp != head){
            if(search_name.compareToIgnoreCase(temp.name) == 0){
                current = temp;
                break;
            }
            temp  = temp.next;
        }
    }
    if(head==null || search_name.compareToIgnoreCase(temp.name) != 0){
        System.out.printf("\n"+ search_name +" is Not Found in Student List..\n\n");
    }
    else
```

Roll No. : 124
Name : Singh Chandani Harendra

Q4. : Write a menu driven program in Java to create a singly circular linked list of student information. Each elements of link list contains information for individual student i.e. (roll no, name marks) provides following operation.
(1) Insert
(2) Search and display (find a node with given name and display all records of list traversing the entire link circularly from searched node.)

```
    {
        System.out.println("Students are : ");
        System.out.printf("\n roll_no \t name \t\t mail_id \t\t marks \n");
        System.out.println(temp.roll_no+"\t\t"+temp.name+"\t\t"+temp.mail_id+"\t\t"+temp.marks);
        temp = temp.next;
        while(temp.name.compareToIgnoreCase(current.name) != 0){
            System.out.println(temp.roll_no+"\t\t"+temp.name+"\t\t"+temp.mail_id+"\t\t"+temp.marks);
            temp=temp.next;
        }
        System.out.println("\n");
    }
}
public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    SinglyCircularLinkedList l = new SinglyCircularLinkedList();
    int roll_no,old_roll_no;
    String name, mail_id;
    int marks;
    char c,ch;
    String s;
    while(true){
        System.out.println("================");
        System.out.println("Options");
        System.out.println("================");
        System.out.println("1 - Insert");
        System.out.println("2 - Search & Display");
        System.out.println("0/<q> - Exit");
        System.out.println("Enter Your Choice : ");
        s = scn.next();
        c = s.charAt(0);
        switch(c){
            case '1':
```

Roll No. : 124
Name : Singh Chandani Harendra

Q4. : Write a menu driven program in Java to create a singly circular linked list of student information. Each elements of link list contains information for individual student i.e. (roll no, name marks) provides following operation.
(1) Insert
(2) Search and display (find a node with given name and display all records of list traversing the entire link circularly from searched node.)

```
                System.out.println("Enter Roll_no : ");
                roll_no = scn.nextInt();
                System.out.println("Enter Name : ");
                name=scn.next();
                System.out.println("Enter Mail_id : ");
                mail_id=scn.next();
                System.out.println("Enter marks : ");
                marks=scn.nextInt();
                l.InsertAtEnd(roll_no, name, mail_id, marks);
                break;
            case '2':
                if(l.isEmpty()){
                    System.out.println("Student List is Empty....");
                    break;
                }
                System.out.println("Enter Name : ");
                name=scn.next();
                l.Search_Display(name);
                break;
            case 'q':
            case '0':
                System.exit(0);
                break;
            default:
                System.out.println("Please enter valid choice....");
                break;
        }
    }
  }
}
```