Roll No. : 124
Name : Singh Chandani Harendra

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
package ds2021;
import java.util.Scanner;
class Array{
    int arr[];
    int size;
    int len;
    Array(int size){
        this.size=size;
        arr = new int[size];
        len = -1;
    }
    Array(){
        this.size=10;
        arr = new int[10];
        len = -1;
    }
    boolean isFull(){
        if(len>=size-1)
            return true;
        else
            return false;
    }
    boolean isEmpty(){
        if(len<0)
            return true;
        else
            return false;
    }
    void InsertAtBegin(int val){
        System.out.println("InsertAtBegin");
        for(int i=len;i>=0;i--){
            arr[i+1]=arr[i];
        }
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
        arr[0]=val;
        len++;
        System.out.println(val +" Inserted..");
    }
    void InsertAtEnd(int val){
        System.out.println("InsertAtEnd");
        len++;
        arr[len] = val;
        System.out.println(val +" Inserted..");
    }
    void InsertAtPos(int val,int pos){
        System.out.println("InsertAtPos");
        if(pos>len+1){
            System.out.println("Position Not..");
        }
        else{
            for(int i=len;i>=pos-1;i--){
                arr[i+1]=arr[i];
            }
            arr[pos-1]=val;
            len++;
            System.out.println(val +" Inserted..");
        }
    }
    void UpdateByVal(int oldval,int newval){
        System.out.println("UpdateByVal");
        int change=0;
        for(int i=0;i<=len;i++){
            if(arr[i]==oldval){
                arr[i]=newval;
                change=1;
            }
        }
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
        if(change==0){
            System.out.println(oldval +" is Not in Array..");
        }
        else{
            System.out.println(oldval +" is Upadated By "+ newval);
        }
    }
    void UpdateByPos(int pos,int newval){
        System.out.println("UpdateByPos");
        if(pos>len+1){
            System.out.println("Position Not..");
        }
        else{
            arr[pos-1]=newval;
            System.out.println("Updated at "+pos);
        }
    }
    void DeleteAtBegin(){
        System.out.println("DeleteAtBegin");
        int delval = arr[0];
        for(int i=0;i<len;i++){
            arr[i]=arr[i+1];
        }
        arr[len]=0;
        len--;
        System.out.println(delval +" Deleted..");
    }
    void DeleteAtEnd(){
        System.out.println("DeleteAtEnd");
        int delval = arr[len];
        arr[len]=0;
        len--;
        System.out.println(delval +" Deleted..");
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
    }
    void DeleteAtPos(int pos){
        System.out.println("DeleteAtPos");
        if(pos>len+1){
            System.out.println("Position Not..");
        }
        else{
            int delval = arr[pos-1];
            for(int i=pos-1;i<len;i++){
                arr[i]=arr[i+1];
            }
            arr[len]=0;
            len--;
            System.out.println(delval +" Deleted..");
        }
    }
    void Display(){
        System.out.println("Array Elements : ");
        for(int i=0;i<=len;i++){
            System.out.println(arr[i]);
        }
    }
    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        System.out.println("Enter Size of Array : ");
        int size = scn.nextInt();
        Array a = new Array(size);
        int pos,val;
        char c,ch;
        String s;
        while(true){
            System.out.println("Options");
            System.out.println("================");
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
System.out.println("1 - Insert");
System.out.println("2 - Update");
System.out.println("3 - Delete");
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':
      if(a.isFull()){
         System.out.println("Array is Full....");
         break;
      }
      System.out.println("Options for Insertion");
      System.out.println("1 - At Begining");
      System.out.println("2 - At Ending");
      System.out.println("3 - At Position");
      System.out.println("Enter  Your Choice : ");
      s = scn.next();
      ch = s.charAt(0);
      switch(ch){
         case '1':
            System.out.println("Enter Value : ");
            val=scn.nextInt();
            a.InsertAtBegin(val);
            break;
         case '2':
            System.out.println("Enter Value : ");
            val=scn.nextInt();
            a.InsertAtEnd(val);
            break;
         case '3':
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
                System.out.println("Enter Value : ");
                val=scn.nextInt();
                System.out.println("Enter Position : ");
                pos=scn.nextInt();
                a.InsertAtPos(val,pos);
                break;
            default :
                System.out.println("Please select valid Choice....");
                break;
        }
        break;
    case '2':
        if(a.isEmpty()){
            System.out.println("Array is Empty....");
            break;
        }
        System.out.println("Options for Upadte");
        System.out.println("1 - By Value");
        System.out.println("2 - By Position");
        System.out.println("Enter Your Choice : ");
        s = scn.next();
        ch = s.charAt(0);
        int newval,oldval;
        switch(ch){
            case '1':
                System.out.println("Enter Old Value : ");
                oldval=scn.nextInt();
                System.out.println("Enter New Value : ");
                newval=scn.nextInt();
                a.UpdateByVal(oldval,newval);
                break;
            case '2':
                System.out.println("Enter Position : ");
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
                pos=scn.nextInt();
                System.out.println("Enter New Value : ");
                newval=scn.nextInt();
                a.UpdateByPos(pos,newval);
                break;
            default :
                System.out.println("Please select valid Choice....");
                break;
        }
        break;
    case '3':
        if(a.isEmpty()){
            System.out.println("Array is Empty....");
            break;
        }
        System.out.println("Options for Deletion");
        System.out.println("1 - At Begining");
        System.out.println("2 - At Ending");
        System.out.println("3 - At Position");
        System.out.println("Enter Your Choice : ");
        s = scn.next();
        ch = s.charAt(0);
        switch(ch){
            case '1':
                a.DeleteAtBegin();
                break;
            case '2':
                a.DeleteAtEnd();
                break;
            case '3':
                System.out.println("Enter Position : ");
                pos=scn.nextInt();
                a.DeleteAtPos(pos);
```

(1) Write a java program to provide the class for Array with all operations on Array data structure. Also provide the menu driven interface to perform that operations.

```
                    break;
                default :
                    System.out.println("Please select valid Choice....");
                    break;
            }
            break;
        case 'p':
        case '4':
            if(a.isEmpty()){
                System.out.println("Array is Empty....");
                break;
            }
            a.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

(2) Write a java program to provide the class for Queue with all operations on Queue data structure. Also provide the menu driven interface to perform that operations.

```
package ds2021;
import java.util.Scanner;
class Queue{
    int queue[];
    int size;
    int front,rear;
    Queue(int size){
        this.size=size;
        queue = new int[size];
        front = rear = -1;
    }
    Queue(){
        this.size=10;
        queue = new int[10];
        front = rear = -1;
    }
    boolean isFull(){
        if(rear>=size-1)
            return true;
        else
            return false;
    }
    boolean isEmpty(){
        if((rear==-1 && front==-1)|| (front>rear) )
            return true;
        else
            return false;
    }
    void Insert(int val){
        if(isFull())
        {
            System.out.println("Queue is Full..");
        }
```

(2) Write a java program to provide the class for Queue with all operations on Queue data structure. Also provide the menu driven interface to perform that operations.

```
    else{
        if(front == -1){
            front++;
        }
        rear++;
        queue[rear]=val;
        System.out.println(val +" Inserted..");
    }
}
void Delete(){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        int val=queue[front];
        queue[front]=0;
        front++;
        System.out.println(val +" Deleted..");
    }
    if(front > rear){
        front = rear = -1;
    }
}
void UpdateByVal(int oldval,int newval){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        int change =0;
        System.out.println("UpdateByVal");
        for(int i=front;i<=rear;i++){
            if(queue[i]==oldval){
                queue[i]=newval;
```

(2) Write a java program to provide the class for Queue with all operations on Queue data structure. Also provide the menu driven interface to perform that operations.

```
                change=1;
            }
        }
        if(change==0){
            System.out.println(oldval +" is Not in Queue..");
        }
        else{
            System.out.println(oldval +" is Upadated By "+ newval);
        }
    }
}
  void UpdateByPos(int pos,int newval){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        System.out.println("UpdateByPos..");
        if(rear+1 < front+pos){
            System.out.println("This Position not ");
        }
        else{
            queue[front+pos-1]=newval;
            System.out.println("Upadated at "+ pos);
        }
    }
}
  void Display(){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        System.out.println("Queue Elements : ");
        for(int i=front;i<=rear;i++){
```

(2) Write a java program to provide the class for Queue with all operations on Queue data structure. Also provide the menu driven interface to perform that operations.

```
            System.out.println("queue[" + i + "] = " + queue[i]);
        }
    }
}
public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    System.out.println("Simple Queue : ");
    System.out.println("Enter Size of Queue : ");
    int size = scn.nextInt();
    Queue q = new Queue(size);
    int pos,val;
    String s;
    char c,ch;
    while(true){
        System.out.println("Options");
        System.out.println("=================");
        System.out.println("1 - Insert (Enqueue)");
        System.out.println("2 - Delete (Dequeue)");
        System.out.println("3 - Update");
        System.out.println("4/<d> - Display");
        System.out.println("0/<q> - Exit");
        System.out.println("Enter Choice : ");
        s = scn.next();
        c = s.charAt(0);
        switch(c){
            case '1':
                System.out.println("Enter Value : ");
                val = scn.nextInt();
                q.Insert(val);
                break;
            case '2':
                q.Delete();
                break;
```

(2) Write a java program to provide the class for Queue with all operations on Queue data structure. Also provide the menu driven interface to perform that operations.

```
case '3':
    if(q.isEmpty()){
        System.out.println("Queue is Empty....");
        break;
    }
    System.out.println("Options for Upadte");
    System.out.println("1 - By Value");
    System.out.println("2 - By Position");
    System.out.println("Enter Choice : ");
    s = scn.next();
    ch = s.charAt(0);
    int newval,oldval;
    switch(ch){
        case '1':
            System.out.println("Enter Old Value : ");
            oldval=scn.nextInt();
            System.out.println("Enter New Value : ");
            newval=scn.nextInt();
            q.UpdateByVal(oldval,newval);
            break;
        case '2':
            System.out.println("Enter Position : ");
            pos=scn.nextInt();
            System.out.println("Enter New Value : ");
            newval=scn.nextInt();
            q.UpdateByPos(pos,newval);
            break;
        default :
            System.out.println("Please select valid Choice....");
            break;
    }
    break;
case 'd':
```

(2) Write a java program to provide the class for Queue with all operations on Queue data structure. Also provide the menu driven interface to perform that operations.

```
            case '4':
                q.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

(3) Write a java program to provide the class for Circular Queue with all operations on Circular Queue data structure. Also provide the menu driven interface to perform that operations.

```
package ds2021;
import java.util.Scanner;
class CQueue{
    int queue[];
    int size;
    int front,rear;
    CQueue(int size){
        this.size=size;
        queue = new int[size];
        front = rear = -1;
    }
    CQueue(){
        this.size=10;
        queue = new int[10];
        front = rear = -1;
    }
    boolean isFull(){
        if((front==0 && rear>=size-1)||(rear==front-1))
            return true;
        else
            return false;
    }
    boolean isEmpty(){
        if(rear==-1 && front==-1)
            return true;
        else
            return false;
    }
    void Insert(int val){
        if(isFull())
        {
            System.out.println("Queue is Full..");
        }
```

(3) Write a java program to provide the class for Circular Queue with all operations on Circular Queue data structure. Also provide the menu driven interface to perform that operations.

```
    else{
        if(front == -1){
            front=rear=0;
        }
        else if(rear>=size-1 && front > 0){
            rear = 0;
        }
        else{
            rear++;
        }
        queue[rear]=val;
        System.out.println(val +" Inserted..");
    }
}
void Delete(){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        int val=queue[front];
        queue[front]=0;
        if(front == rear){
            front = rear = -1;
        }
        else if(front >= size-1 && rear >= 0)
        {
            front = 0;
        }
        else{
            front++;
        }
        System.out.println(val +" Deleted..");
    }
```

(3) Write a java program to provide the class for Circular Queue with all operations on Circular Queue data structure. Also provide the menu driven interface to perform that operations.

```
}
void UpdateByVal(int oldval,int newval){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        System.out.println("UpdateByVal");
        int change =0;
        for(int i=0;i<size;i++){
            if(queue[i]==oldval){
                queue[i]=newval;
                change =1;
            }
        }
        if(change==0){
            System.out.println(oldval +" is Not in Queue..");
        }
        else{
            System.out.println(oldval +" is Upadated By "+ newval);
        }
    }
}
void Display(){
    if(isEmpty()){
        System.out.println("Queue is Empty..");
    }
    else{
        System.out.println("Queue Elements : ");
        for(int i=0;i<size;i++){
            System.out.println("queue[" + i + "] = " + queue[i]);
        }
    }
}
```

(3) Write a java program to provide the class for Circular Queue with all operations on Circular Queue data structure. Also provide the menu driven interface to perform that operations.

```
public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    System.out.println("Circular Queue : ");
    System.out.println("Enter Size of Queue : ");
    int size = scn.nextInt();
    CQueue q = new CQueue(size);
    int val;
    String s;
    char c;
    while(true){
        System.out.println("Options");
        System.out.println("================");
        System.out.println("1 - Insert (Enqueue)");
        System.out.println("2 - Delete (Dequeue)");
        System.out.println("3 - Update");
        System.out.println("4/<d> - Display");
        System.out.println("0/<q> - Exit");
        System.out.println("Enter Choice : ");
        s = scn.next();
        c = s.charAt(0);
        switch(c){
            case '1':
                System.out.println("Enter Value : ");
                val = scn.nextInt();
                q.Insert(val);
                break;
            case '2':
                q.Delete();
                break;
            case '3':
                System.out.println("Enter old Value : ");
                int oldval = scn.nextInt();
                System.out.println("Enter New Value : ");
```

(3) Write a java program to provide the class for Circular Queue with all operations on Circular Queue data structure. Also provide the menu driven interface to perform that operations.

```
                val = scn.nextInt();
                q.UpdateByVal(oldval,val);
                break;
            case 'd':
            case '4':
                q.Display();
                break;
            case 'q':
            case '0':
                System.exit(0);
                break;
            default:
                System.out.println("Please enter valid choice....");
                break;
        }
    }
  }
}
```