**QUEUES**

**Q1.**

1 import java.util.\*;  
 2 public class Queue\_Arr{  
 3 static int front = -1;  
 4 static int rear = -1;  
 5 public static void main (String args[]){  
 6 Scanner sc = new Scanner(System.in);  
 7 System.out.print("\nEnter the size of queue : ");  
 8 int size = sc.nextInt();  
 9 int[] arr = new int[size];  
10 while(true){  
11 System.out.print("\nPress 1 to Enter element into QUEUE");  
12 System.out.print("\nPress 2 to Delete element from QUEUE");  
13 System.out.print("\nPress 3 to Display elements of QUEUE");  
14 System.out.print("\nPress 4 to Exit\n");  
15 int choice = sc.nextInt();  
16 if(choice==1){  
17 System.out.print("\nEnqueue element, Please enter the element : ");  
18 int element = sc.nextInt();  
19 enqueue(arr,element,size);  
20 }  
21 else if(choice == 2){  
22 System.out.print("\nDequeue element");  
23 dequeue(arr,size);  
24 }  
25 else if (choice == 3){  
26 System.out.print("\nDisplaying elements");  
27 display(arr);  
28 }  
29 else if (choice == 4){  
30 System.out.print("\nExitting program");  
31 System.exit(0);  
32 }  
33 else{  
34 System.out.print("\nInvalid Input! Enter again.");  
35 }  
36 }  
37 }  
38   
39 public static void enqueue(int[] arr, int element, int size){  
40 if(rear+1<size){  
41 arr[rear+1]=element;  
42 rear=rear+1;  
43 }  
44 else if(rear+1==size){  
45 System.out.print("\nQueue is FULL");  
46 }  
47 }  
48   
49 public static void dequeue(int [] arr, int size){  
50 if(rear==-1){  
51 System.out.print("\nQueue is EMPTY");  
52 }  
53 else{  
54 for(int i = 0; i<size-1; i++){  
55 arr[i] = arr[i+1];  
56 }  
57 rear = rear-1;  
58 }  
59 }  
60   
61 public static void display(int [] arr){  
62 if(rear==-1){  
63 System.out.print("\nQueue is EMPTY");  
64 }  
65   
66 System.out.print("\n");  
67 for (int i = 0; i<=rear ; i++){  
68 System.out.print(arr[i]+" ");  
69 }  
70 }  
71 }

**Q2.**

1 import java.util.\*;  
 2 class Node{  
 3 int data;  
 4 Node next;  
 5 Node(){  
 6 this.data = 0;  
 7 this.next = null;  
 8 }  
 9 }  
 10 public class Queue\_Node{  
 11 static Node head;  
 12 public static void main(String args[]){  
 13 Scanner sc = new Scanner(System.in);  
 14 while(true){  
 15 System.out.print("\nPress 1 to Enter element into QUEUE");  
 16 System.out.print("\nPress 2 to Delete element from QUEUE");  
 17 System.out.print("\nPress 3 to Display elements of QUEUE");  
 18 System.out.print("\nPress 4 to Display even elements of QUEUE");  
 19 System.out.print("\nPress 5 to Exit\n");  
 20 int choice = sc.nextInt();  
 21 if(choice==1){  
 22 System.out.print("\nEnqueue element, Please enter the element : ");  
 23 int element = sc.nextInt();  
 24 enqueue(element);  
 25 }  
 26 else if(choice == 2){  
 27 System.out.print("\nDequeue element");  
 28 dequeue();  
 29 }  
 30 else if (choice == 3){  
 31 System.out.print("\nDisplaying elements :");  
 32 print\_queue();  
 33 }  
 34 else if(choice==4){  
 35 System.out.print("\nDisplaying even elements :");  
 36 even\_queue();  
 37 }  
 38 else if (choice == 5){  
 39 System.out.print("\nExitting program");  
 40 System.exit(0);  
 41 }  
 42 else{  
 43 System.out.print("\nInvalid Input! Enter again.");  
 44 }  
 45 }  
 46   
 47 }  
 48   
 49 public static void enqueue(int data\_){  
 50 Node new\_node = new Node();  
 51 new\_node.data = data\_;  
 52 Node temp = new Node();  
 53 temp = head;  
 54 if(head==null){  
 55 head = new\_node;  
 56 }  
 57 else{  
 58 while(temp.next!=null){  
 59 temp = temp.next;  
 60 }  
 61 temp.next = new\_node;   
 62 }   
 63 }  
 64   
 65 public static void dequeue(){  
 66 Node temp = new Node();  
 67 temp = head;  
 68 if(head!= null){  
 69 head = temp.next;  
 70 temp.next = null;  
 71 }  
 72 else  
 73 System.out.print("\nQueue in empty");   
 74 }  
 75   
 76 public static void even\_queue(){  
 77 Node temp = new Node();  
 78 temp = head;  
 79 int count =0;  
 80 while(temp!=null){  
 81 if(temp.data%2!=0){  
 82 count=count+1;  
 83 }  
 84 temp = temp.next;  
 85   
 86 }  
 87 System.out.print("\nThe count of even numbers is :"+count);  
 88 }  
 89   
 90 public static void print\_queue(){  
 91 Node temp = new Node();  
 92 temp = head;  
 93 if(temp == null){  
 94 System.out.print("\nNo Elements found!\n");  
 95 }  
 96 else{  
 97 while(temp.next!=null){  
 98 System.out.print(temp.data+"--->");  
 99 temp = temp.next;  
100 }  
101 System.out.print(temp.data+"\n");  
102 }  
103 }  
104   
105   
106   
107 }  
108

**Q3.**

**ALREADY DONE IN Q2.**

**Q5.**

1 import java.util.\*;  
 2 class Queue\_A5{  
 3 static int top1 = -1;  
 4 static int top2 = -1;  
 5 public static void main(String args[]){  
 6 Scanner sc = new Scanner(System.in);  
 7 int[] array = new int[100];  
 8 int choice = 0;  
 9 do{  
10 System.out.println(" Enter 1 to enqueue element in queue.\n Enter 2 to print queue.\n Enter 3 to exit.");  
11 choice = sc.nextInt();  
12 if(choice==1){  
13 System.out.print("Enter an element: ");  
14 int a = sc.nextInt();  
15 enqueue(1,a,array);  
16 }  
17 else if(choice==3) System.out.println("Program terminated.");  
18 else if(choice==2){  
19 print(array);  
20 }  
21 else System.out.println("Invalid Input, Enter again ");  
22 }while(choice!=3);  
23 }  
24 public static void enqueue(int b,int a,int array[]){  
25 if(b ==1){  
26 top1++;  
27 array[top1] = a;  
28 }  
29 if(b ==2){  
30 top2++;  
31 array[top2] = a;  
32 }  
33 }  
34 public static void print(int array[]){  
35 if(top1==-1) System.out.println("Empty Queue.");  
36 else{  
37 int[] array2 = new int[array.length];  
38 for(int i = top1; i>=0; i--){  
39 enqueue(2,array[i],array2);  
40 }  
41 while(top2>-1){  
42 System.out.print(array2[top2]+" ");  
43 top2--;  
44 }  
45 System.out.println();  
46 }  
47 }   
48 }  
49

**Q6.**

1 import java.util.\*;  
 2 class Queue\_A6{  
 3 static int last = -1;  
 4 static int first = -1;  
 5 public static void main(String args[]){  
 6 Scanner sc = new Scanner(System.in);  
 7 String[] array = new String[100];  
 8 System.out.print(" Enter the string : ");  
 9 String st = sc.nextLine();  
10 String[] str = st.split(" ");  
11 System.out.println("Entered string");  
12 System.out.println(st);  
13 for(String i : str){  
14 enqueue(i,array);  
15 }  
16 reverse(array);  
17 }  
18 public static void enqueue(String a,String array[]){  
19 if(last == -1){  
20 last ++;  
21 first ++;  
22 array[first] = a;  
23 }  
24 else{  
25 last++;  
26 array[last] = a;  
27 }  
28 }  
29 public static void reverse(String array[]){  
30 String result = "";  
31 if(first==-1) System.out.println("Empty Queue.");  
32 else{  
33 for(int i = first; i<=last; i++){  
34 result = dequeue(array)+" "+result;  
35 }  
36 }  
37 System.out.println("Reversed String");  
38 System.out.println(result);  
39 }   
40 public static String dequeue(String array[]){  
41 if(first == last){  
42 String a = array[first];  
43 first=-1;  
44 last = -1;  
45 return a;  
46 }  
47 else{  
48 String a = array[first];  
49 first++;  
50 return a;  
51 }  
52 }  
53 }

**SET B**

**Q1.**

1 import java.util.\*;  
 2 class CircularQueue{  
 3 static int last = -1;  
 4 static int first = -1;  
 5 public static void main(String args[]){  
 6 Scanner sc = new Scanner(System.in);  
 7 int[] array = new int[5];  
 8 int choice = 0;  
 9 do{  
 10 System.out.println(" Enter 1 to enqueue element in queue.\n Enter 2 to dequeue element.\n Enter 3 to print first element.\n Enter 4 to print queue.\n Enter 5 to exit.");  
 11 choice = sc.nextInt();  
 12 if(choice==1){  
 13 System.out.print("Enter an element: ");  
 14 int a = sc.nextInt();  
 15 enqueue(a,5,array);  
 16 }  
 17 else if(choice==5) System.out.println("Program terminated.");  
 18 else if(choice==4){  
 19 print(5,array);  
 20 }  
 21 else if(choice==2){  
 22 dequeue(5,array);  
 23 }  
 24 else if(choice==3){  
 25 top(array);  
 26 }  
 27 else System.out.println("Invalid Input, Enter again ");  
 28 }while(choice!=5);  
 29 }  
 30 public static void enqueue(int a, int size,int array[]){  
 31 if((first == 0 && last == size-1) || (last == first-1)){  
 32 System.out.println("Queue full!!!!!!");  
 33 }  
 34 else{  
 35 if(last == -1 && first == -1){  
 36 last ++;  
 37 first ++;  
 38 array[first] = a;  
 39 }  
 40 else if(last < size-1 && last >= first){  
 41 last++;  
 42 array[last] = a;  
 43 }  
 44 else if(first >0){  
 45 if(last == size-1){  
 46 last = 0;  
 47 array[last]=a;  
 48 }  
 49 else{  
 50 last++;  
 51 array[last] = a;  
 52 }  
 53 }  
 54 }  
 55 }  
 56 public static void dequeue(int size,int array[]){  
 57 if(first == last){  
 58 if(first !=-1){  
 59 System.out.println(array[first]);  
 60 first=-1;  
 61 last = -1;  
 62 }  
 63 else System.out.println("Queue Empty!!!");  
 64 }  
 65 else if((first != last) && (first == size-1)){  
 66 System.out.println(array[first]);  
 67 first=0;  
 68 }  
 69 else{  
 70 System.out.println(array[first]);  
 71 first++;  
 72 }  
 73 }  
 74 public static void top(int array[]){  
 75 if(first==-1){  
 76 System.out.println("Queue Empty!!!");  
 77 }  
 78 else{  
 79 System.out.println(array[first]);  
 80 }  
 81 }  
 82 public static void print(int size,int array[]){  
 83 if(first==-1) System.out.println("Empty Queue.");  
 84 else if(last >= first){  
 85 for(int i = first; i<=last; i++){  
 86 System.out.print(array[i]+" ");  
 87 }  
 88 System.out.println();  
 89 }  
 90 else{  
 91 for(int i = first; i<size; i++){  
 92 System.out.print(array[i]+" ");  
 93 }  
 94 for(int i = 0; i<=last; i++){  
 95 System.out.print(array[i]+" ");  
 96 }  
 97 System.out.println();  
 98 }  
 99 }   
100 }  
101

**Q2.**

1 import java.util.\*;  
 2 class Queue\_B2{  
 3 static int last = -1;  
 4 static int first = -1;  
 5 public static void main(String args[]){  
 6 Scanner sc = new Scanner(System.in);  
 7 int[] array = new int[5];  
 8 int choice = 0;  
 9 do{  
 10 System.out.println(" Enter 1 to enqueue element in queue.\n Enter 2 to dequeue element.\n Enter 3 to print queue.\n Enter 4 to count number of duplicates in queue.\n Enter 5 to exit.");  
 11 choice = sc.nextInt();  
 12 if(choice==1){  
 13 System.out.print("Enter an element: ");  
 14 int a = sc.nextInt();  
 15 enqueue(a,5,array);  
 16 }  
 17 else if(choice==5) System.out.println("Program terminated.");  
 18 else if(choice==4){  
 19 dupli(5,array);  
 20 }  
 21 else if(choice==2){  
 22 dequeue(5,array);  
 23 }  
 24 else if(choice==3){  
 25 print(5,array);  
 26 }  
 27 else System.out.println("Invalid Input, Enter again ");  
 28 }while(choice!=5);  
 29   
 30 }  
 31 public static void enqueue(int a, int size,int array[]){  
 32 if((first == 0 && last == size-1) || (last == first-1)){  
 33 System.out.println("Queue full!!!!!!");  
 34 }  
 35 else{  
 36 if(last == -1 && first == -1){  
 37 last ++;  
 38 first ++;  
 39 array[first] = a;  
 40 }  
 41 else if(last < size-1 && last >= first){  
 42 last++;  
 43 array[last] = a;  
 44 }  
 45 else if(first >0){  
 46 if(last == size-1){  
 47 last = 0;  
 48 array[last]=a;  
 49 }  
 50 else{  
 51 last++;  
 52 array[last] = a;  
 53 }  
 54 }  
 55 }  
 56 }  
 57 public static void dequeue(int size,int array[]){  
 58 if(first == last){  
 59 if(first !=-1){  
 60 System.out.println(array[first]);  
 61 first=-1;  
 62 last = -1;  
 63 }  
 64 else System.out.println("Queue Empty!!!");  
 65 }  
 66 else if((first != last) && (first == size-1)){  
 67 System.out.println(array[first]);  
 68 first=0;  
 69 }  
 70 else{  
 71 System.out.println(array[first]);  
 72 first++;  
 73 }  
 74 }  
 75 public static void print(int size,int array[]){  
 76 if(first==-1) System.out.println("Empty Queue.");  
 77 else if(last >= first){  
 78 for(int i = first; i<=last; i++){  
 79 System.out.print(array[i]+" ");  
 80 }  
 81 System.out.println();  
 82 }  
 83 else{  
 84 for(int i = first; i<size; i++){  
 85 System.out.print(array[i]+" ");  
 86 }  
 87 for(int i = 0; i<=last; i++){  
 88 System.out.print(array[i]+" ");  
 89 }  
 90 System.out.println();  
 91 }  
 92 }   
 93 public static void dupli(int size,int array[]){  
 94 if(first==-1) System.out.println("Empty Queue.");  
 95 else if(last >= first){  
 96 int count = 0;  
 97 for(int i=first; i<=last; i++){  
 98 for(int j=i+1; j<=last; j++){  
 99 if(array[i] == array[j]){  
100 count++;  
101 break;  
102 }  
103 }  
104 }  
105 System.out.println("Total number of duplicate elements in queue : "+count);  
106 }  
107 else{  
108 int count = 0;  
109 for(int i = first; i<size; i++){  
110 for(int j=i+1; j<size; j++){  
111 if(array[i] == array[j]){  
112 count++;  
113 break;  
114 }  
115 }  
116 }  
117 for(int i = 0; i<=last; i++){  
118 for(int j=i+1; j<=last; j++){  
119 if(array[i] == array[j]){  
120 count++;  
121 break;  
122 }  
123 }  
124 }  
125 System.out.println("Total number of duplicate elements in queue : "+count);  
126 }  
127 }   
128 }  
129

**Q3 Priority Queue**

1 import java.util.\*;  
 2 class Last{  
 3 int data;  
 4 Last next ;  
 5 int prio ;  
 6 Last(){  
 7 data = 0;  
 8 next = null;  
 9 prio = 0;  
 10 }  
 11 }  
 12 public class PriorityQueueLinkedList{  
 13 static Last last;  
 14 static Last first;  
 15 public static void main(String args[]){  
 16 Scanner sc = new Scanner(System.in);  
 17 int choice = 0;  
 18 do{  
 19 System.out.println(" Enter 1 to enqueue element in queue.\n Enter 2 to dequeue element.\n Enter 3 to print first element.\n Enter 4 to print queue.\n Enter 5 to exit.");  
 20 choice = sc.nextInt();  
 21 if(choice==1){  
 22 System.out.print("Enter an element: ");  
 23 int a = sc.nextInt();  
 24 System.out.print("Enter priority number: ");  
 25 int b = sc.nextInt();  
 26 enqueue(a,b);  
 27 }  
 28 else if(choice==5) System.out.println("Program terminated.");  
 29 else if(choice==4){  
 30 print();  
 31 }  
 32 else if(choice==2){  
 33 dequeue();  
 34 }  
 35 else if(choice==3){  
 36 top();  
 37 }  
 38 else System.out.println("Invalid Input, Enter again ");  
 39 }while(choice!=5);  
 40 }  
 41 public static void enqueue(int a, int b){  
 42 Last n = new Last();  
 43 n.data = a;  
 44 n.prio = b;  
 45 n.next = null;  
 46 if(first == null){  
 47 first = n;  
 48 last = n;  
 49 }  
 50 else if(first != null && b <first.prio){  
 51 if(first.next==null){  
 52 n.next = last;  
 53 first = n;  
 54 }  
 55 else{  
 56 n.next = first;  
 57 first = n;  
 58 }  
 59 }  
 60 else if(first.next == null && b>=first.prio){  
 61 last = n;  
 62 first.next = last;  
 63 }  
 64 else{  
 65 Last temp = first;  
 66 while(b>=temp.next.prio && temp.next!= null){  
 67 temp = temp.next;  
 68 if(temp.next == null) break;  
 69 }  
 70 if(temp.next == null){  
 71 last.next = n;  
 72 last = n;  
 73 }  
 74 else{  
 75 n.next = temp.next;  
 76 temp.next = n;  
 77 }  
 78 }  
 79 }  
 80 public static void print(){  
 81 Last temp = new Last();  
 82 temp = first;  
 83 if(first == null) System.out.println("Queue empty");  
 84 else{  
 85 while(temp!=last){  
 86 System.out.print(temp.data+" ");  
 87 temp = temp.next;  
 88 }  
 89 System.out.print(temp.data);  
 90 System.out.println();  
 91 }  
 92 }  
 93 public static void dequeue(){  
 94 Last temp = new Last();  
 95 if(first==null) System.out.println("Queue Empty");  
 96 else if(first == last){  
 97 System.out.println("Dequeued element : "+first.data);  
 98 first = null;  
 99 last = null;  
100 }  
101 else{  
102 System.out.println("Dequeued element : "+first.data);  
103 first = first.next;  
104 }  
105 }  
106 public static void top(){  
107 if(first==null) System.out.println("Queue Empty");  
108 else System.out.println("first element : "+first.data);  
109 }  
110 }

**BINAY 19CSU370**

**IOT-A**