



NAME : PRIYADARSHAN GHOSH

COLLEGE ROLL NO: 72

UNIVERSITY ROLL NO: 16900319072

DEPARTMENT: ECE-1(Y)

SEMESTER: 3rd

PAPER CODE : ES-CS391

➤ **Laboratory Assignment #6**

Write menu driven program to perform following operations using functions:

A.IMPLEMENTATION OF LINEAR QUEUE OPERATIONS LIKE ENQUEUE(), DEQUEUE() AND DISPLAYQ() USING ARRAY.

Ans:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 50
int LQ[MAX];
int rear = -1;
int front = -1;

void enQueue(int x) {

    if(rear == MAX-1){
        printf("Queue is full");
    }
    if(rear == -1 && front == -1){
        front = rear = 0;
    }
    else
        rear++;
    LQ[rear] = x;
}
```

```
int deQueue() {  
    int x;  
    if(front == -1 || front > rear) {  
        printf("Queue is empty");  
        return -1;  
    }  
    x = LQ[front++];  
    return x;  
}
```

```
void displayQ() {  
    int i;  
    for(i=front; i<= rear; i++)  
        printf("%d ", LQ[i]);  
}
```

```
int main() {  
    int x, p;  
    while(1){  
        printf("\n Press 1 to insert an element");  
        printf("\n Press 2 to delete an element");  
        printf("\n Press 3 to display elements");  
        printf("\n Press 4 to exit");  
        printf("\n ENTER THE OPERATION : ");  
        scanf("%d",&x);  
        switch(x){  
            case 1: printf("\n Enter an element to insert:");
```

```

scanf("%d",&p);
enQueue(p);
break;
case 2: p = deQueue();
        if(front >= -1 && p!=-1)
            printf("\n The deleted element is %d",p);
        break;
case 3: displayQ();
        break;
case 4: exit(0);
    }
}
}

```

OUTPUT =>

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:10

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:20

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:30

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:40

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:50

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 3

10 20 30 40 50

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 2

The deleted element is 10

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 3

20 30 40 50

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 4

Process exited after 43.74 seconds with return value 0

Press any key to continue . . .

B.IMPLEMENTATION OF LINEAR QUEUE OPERATIONS LIKE ENQUEUE(), DEQUEUE() AND DISPLAYQ() USING STRUCTURE .

Ans:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 50
struct lq{
    int queue[MAX];
    int rear;
    int front;
};
struct lq Q;

void enQueue(int x) {

    if(Q.rear == MAX-1){
        printf("Queue is full");
    }
```

```
if(Q.rear == -1 && Q.front == -1){
    Q.front = Q.rear = 0;
}
else
    Q.rear++;
Q.queue[Q.rear] = x;
}

int deQueue() {
    int x;
    if(Q.front == -1 || Q.front > Q.rear) {
        printf(" Queue is empty");
        return -1;
    }
    x = Q.queue[Q.front++];
    return x;
}

void displayQ() {
    int i;
    for(i=Q.front; i<= Q.rear; i++)
        printf("%d ", Q.queue[i]);
}

int main() {
    int x, p;
    Q.front=-1;
```


Q.rear=-1;

while(1){

printf("\n Press 1 to insert an element");

printf("\n Press 2 to delete an element");

printf("\n Press 3 to display elements");

printf("\n Press 4 to exit");

printf("\n ENTER THE OPERATION : ");

scanf("%d",&x);

switch(x){

case 1: printf("\n Enter an element to insert:");

scanf("%d",&p);

enQueue(p);

break;

case 2: p = deQueue();

if(Q.front >= -1 && p!=-1)

printf("\n The deleted element is %d",p);

break;

case 3: displayQ();

break;

case 4: exit(0);

}

}

}

OUTPUT =>

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:11

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:22

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:33

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:44

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:55

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 3

11 22 33 44 55

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 2

The deleted element is 11

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 3

22 33 44 55

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 4

Process exited after 20.32 seconds with return value 0

Press any key to continue . . .

C.IMPLEMENTATION OF LINEAR QUEUE OPERATIONS LIKE ENQUEUE(), DEQUEUE() AND DISPLAYQ() USING STRUCTURE POINTERS.

Ans:

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#define MAX 50
```

```
struct lq{
```

```
    int queue[MAX];
```

```
    int rear;
    int front;
};

void enqueue(struct lq *Q,int x) {

    if(Q->rear == MAX-1){
        printf("Queue is full");
    }
    if(Q->rear == -1 && Q->front == -1){
        Q->front = Q->rear = 0;
    }
    else
        Q->rear++;
    Q->queue[Q->rear] = x;
}

int dequeue(struct lq *Q) {
    int x;
    if(Q->front == -1 || Q->front > Q->rear) {
        printf(" Queue is empty");
        return -1;
    }
    x = Q->queue[Q->front++];
    return x;
}
```

```
void displayQ(struct lq *Q) {  
    int i;  
    for(i=Q->front; i<= Q->rear; i++)  
        printf("%d ", Q->queue[i]);  
}
```

```
int main() {  
    int x, p;  
    struct lq *Q;  
    Q = (struct lq *)malloc(sizeof(struct lq));  
    Q->front=-1;  
    Q->rear=-1;  
    while(1){  
        printf("\n Press 1 to insert an element");  
        printf("\n Press 2 to delete an element");  
        printf("\n Press 3 to display elements");  
        printf("\n Press 4 to exit");  
        printf("\n ENTER THE OPERATION : ");  
        scanf("%d",&x);  
        switch(x){  
            case 1: printf("\n Enter an element to insert:");  
                    scanf("%d",&p);  
                    enqueue(Q,p);  
                    break;  
            case 2: p = dequeue(Q);  
                    if(Q->front >= -1 && p!=-1)
```

```

                                printf("\n The deleted element is %d",p);
                                break;
                        case 3: displayQ(Q);
                                break;
                        case 4: exit(0);
                }
        }
}

```

OUTPUT =>

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:15

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:25

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:35

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:45

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 1

Enter an element to insert:55

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 3

15 25 35 45 55

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 2

The deleted element is 15

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 3

25 35 45 55

Press 1 to insert an element

Press 2 to delete an element

Press 3 to display elements

Press 4 to exit

ENTER THE OPERATION : 4

Process exited after 30.18 seconds with return value 0

Press any key to continue . . .