

## Assignment :: 4

Name ::- Abhishek Kumar

Regd.No ::- 190101120066

Branch :: B.tech(CSE)

**Q.1.** Write a menu driven program to implement queue operations such as Insert, Delete, Display, whether queue is empty etc by using array.

```
Q.1. #include <stdio.h>
#define MAX 10

int queue = arr[MAX];
int rear = -1;
int front = -1;

void insert(int item);
int del();
int peek();
void display();
int isFull();
int isEmpty();

int main()
{
    int choice, item;
    while(1)
    {
        printf("\n1. Insert\n");
        printf("\n2. Delete\n");
        printf("\n3. Display elements of the queue\n");
        printf("\n4. Display all elements of the queue\n");
        printf("\n5. Quit\n");
        printf("\nEnter your choice:");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1:
                printf("\nEnter the elements of adding in queue:");
                scanf("%d", &item);
                insert(item);
                break;
            case 2:
                item = del();
                printf("\nDeleted element is: %d\n", item);
                break;
            case 3:
                printf("\nElement of the front is: %d\n", peek());
                break;
            case 4:
                display();
                break;
            case 5:
                exit(1);
        }
    }
}
```

```
switch(choice)
{
    case 1:
        printf("\nEnter the elements of adding in queue:");
        scanf("%d", &item);
        insert(item);
        break;
    case 2:
        item = del();
        printf("\nDeleted element is: %d\n", item);
        break;
    case 3:
        printf("\nElement of the front is: %d\n", peek());
        break;
    case 4:
        display();
        break;
    case 5:
        exit(1);
}
```

```
printf("\nExiting choice\n");
// End of switch
// End of while

return 0;
// End of main()
void insert(int item)
{
    if(isFull())
    {
        printf("\nQueue overflow\n");
        return;
    }
    if(front == -1)
        front = 0;
    rear = rear + 1;
    queue[rear] = item;
    // End of insert()
}

int del()
{
    int item;
    if(isEmpty())
    {
        printf("\nQueue underflow\n");
        return -1;
    }
    item = queue[front];
    front = front + 1;
    return item;
}
```

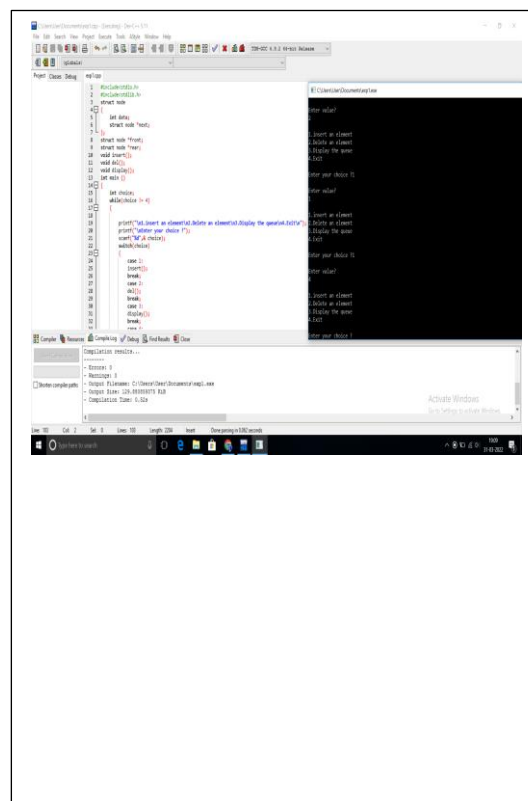
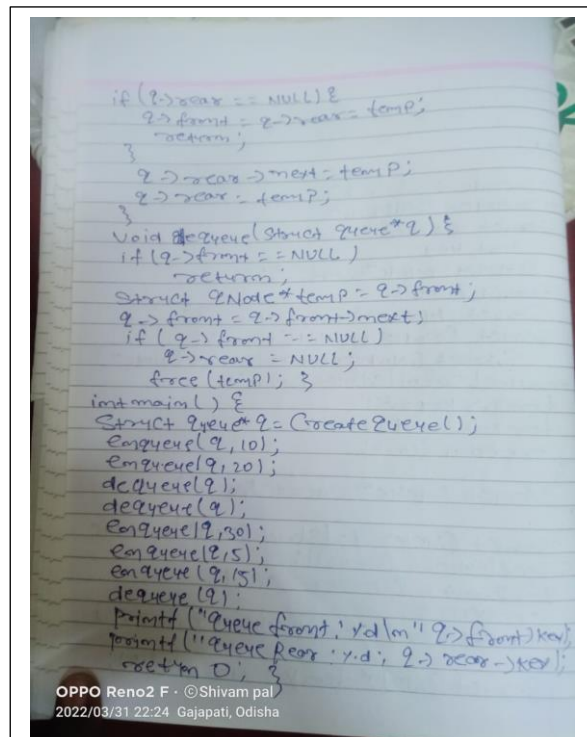
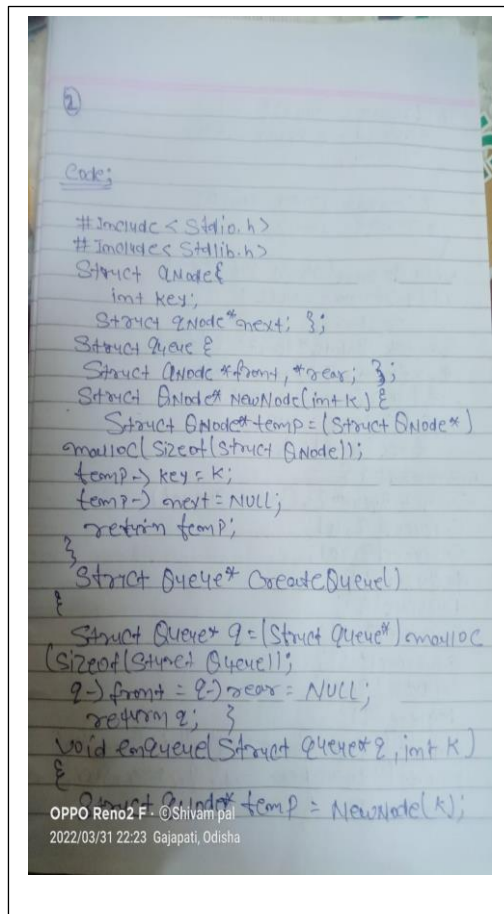
```
int isEmpty()
{
    if(front == -1 || front == rear + 1)
        return 1;
    else
        return 0;
}

// End of isEmpty()

int isFull()
{
    if(rear == MAX - 1)
        return 1;
    else
        return 0;
}

// End of isFull()
void display()
{
    int i;
    if(isEmpty())
    {
        printf("\nQueue is empty\n");
        return;
    }
    printf("\nQueue is: \n");
    for(i = front; i <= rear; i++)
        printf("%d ", queue[i]);
    printf("\n");
}
```

**Q.2.** Write a menu driven program to implement queue operations such as Insert, Delete, Display, whether queue is empty etc by using linked list.





#### Q.4. WAP to implement the double ended queue using array.

```
scanf("%d", &Q);
switch(Q)
{
    case 1: printf("Enter number of elements:");
            scanf("%d", &n);
            initialize(&Q);
            printf("Enter the data:");
            for(i=0; i<n; i++)
            {
                scanf("%d", &x);
                if(full(&Q))
                {
                    printf("Queue is full!!");
                    exit(0);
                }
                enqueue(&Q, x);
            }
            break;
    case 2: printf("Enter element to be inserted:");
            scanf("%d", &x);
            if(full(&Q))
            {
                printf("Queue is full!!");
                exit(0);
            }
            enqueue(&Q, x);
}
```

OPPO Reno2 F, ©Shivam pal  
2022/03/31 22:38 Gajapati, Odisha

```
scanf("%d", &Q);
switch(Q)
{
    case 1: printf("Enter number of elements:");
            scanf("%d", &n);
            initialize(&Q);
            printf("Enter the data:");
            for(i=0; i<n; i++)
            {
                scanf("%d", &x);
                if(full(&Q))
                {
                    printf("Queue is full!!");
                    exit(0);
                }
                enqueue(&Q, x);
            }
            break;
    case 2: printf("Enter element to be inserted:");
            scanf("%d", &x);
            if(full(&Q))
            {
                printf("Queue is full!!");
                exit(0);
            }
            enqueue(&Q, x);
}
```

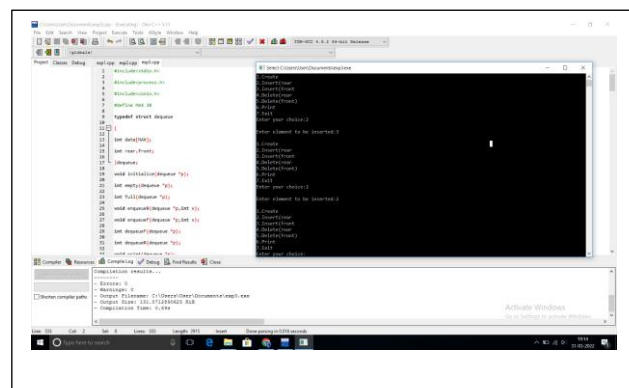
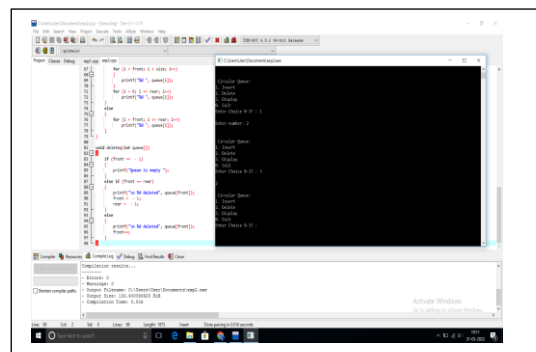
OPPO Reno2 F, ©Shivam pal  
2022/03/31 22:38 Gajapati, Odisha

```
if(p>rear == p>front) // delete the last
    element;
initialize(p);
else
    p>rear = (p>rear+1)% MAX;
return(x);
}

int dequeue(dequeue* p)
{
    int x;
    x = p->data[p->rear];
    if(p->rear == p->front)
        initialize(p);
    else
        p->rear = (p->rear+1)% MAX;
    return(x);
}

void print(dequeue* p)
{
    if(empty(p))
    {
        printf("Queue is empty!!");
        exit(0);
    }
    int i;
    i = p->front;
    while(i != p->rear)
    {
        printf("%d ", p->data[i]);
        i = (i+1)% MAX;
    }
}
```

OPPO Reno2 F, ©Shivam pal  
2022/03/31 22:39 Gajapati, Odisha





```

Case 3: printf("\n Enter the element to be
inserted:");
scanf("%d", &x);
if (full(q))
{
    printf("\n Queue is full!!");
    exit(0);
}
enqueue(q, x);
break;
Case 4: if (empty(q))
{
    printf("\n Queue is empty!!");
    exit(0);
}
x = dequeue(q);
printf("\n Element deleted is %d\n", x);
break;
Case 5: if (empty(q))
{
    printf("\n Queue is empty!!");
    exit(0);
}
x = dequeue(q);
printf("\n Element deleted is %d\n", x);
break;

```

OPPO Reno2 F • ©Shivam pal  
2022/03/31 22:38 Gajapati, Odisha

```

Case 6: printf("%d", q);
break;
default: break;
}
while (op != 7);
return 0;
}
void initialize(dequeue *p)
{
    p->rear = -1;
    p->front = -1;
}
int empty(dequeue *p)
{
    if (p->rear == -1)
        return 1;
    return 0;
}
int full(dequeue *p)
{
    if ((p->rear + 1) % MAX == p->front)
        return 1;
    return 0;
}
void enqueue(dequeue *p, int x)
{
    // ...
}

```

OPPO Reno2 F • ©Shivam pal  
2022/03/31 22:38 Gajapati, Odisha

```

scanf("%d", &op);
switch(op)
{
    Case 1: printf("\n Enter number of elements:");
        scanf("%d", &n);
        initialize(q);
        printf("\n Enter the data:");
        for (i = 0; i < n; i++)
        {
            scanf("%d", &x);
            if (full(q))
            {
                printf("\n Queue is full!!");
                exit(0);
            }
            enqueue(q, x);
        }
        break;
    Case 2: printf("\n Enter element to be inserted:");
        scanf("%d", &x);
        if (full(q))
        {
            printf("\n Queue is full!!");
            exit(0);
        }
        enqueue(q, x);
}

```

OPPO Reno2 F • ©Shivam pal  
2022/03/31 22:38 Gajapati, Odisha

```

Case 3: printf("\n Enter the element to be
inserted:");
scanf("%d", &x);
if (full(q))
{
    printf("\n Queue is full!!");
    exit(0);
}
enqueue(q, x);
break;
Case 4: if (empty(q))
{
    printf("\n Queue is empty!!");
    exit(0);
}
x = dequeue(q);
printf("\n Element deleted is %d\n", x);
break;
Case 5: if (empty(q))
{
    printf("\n Queue is empty!!");
    exit(0);
}
x = dequeue(q);
printf("\n Element deleted is %d\n", x);
break;

```

OPPO Reno2 F • ©Shivam pal  
2022/03/31 22:38 Gajapati, Odisha