1) Implement the linear queue
2) Implement the circular queue

1) Implement the linear queue

```
#include<stdio.h>

//#include<conio.h>

#include<stdlib.h>

#define size 15

int queue[size],front=0,rear=0,data;

int res;

void enqueue();

void dequeue();

void display();

int main()

{

	int c;

//	clrscr();

	printf("\n1.Insertion\n2.Deletion\n3.Display");

	do

	{

		printf("\n\nEnter your Choice :: ");

		scanf("%d",&c);

		switch(c)

		{

			case 1:

			enqueue();

			break;

			case 2:

			dequeue();
```

```
                    break;
                    case 3:
                    printf("\n\nContents of queue is \t");
                    display();
                    break;
                    default:
                    printf("\nInvalid Choice......");
                    exit(0);
                }
        }while(c<4);
//      getch();
}
void enqueue()
{
        if(rear>=size)
        {
                printf("\nOverflow");
                return;
        }
        else
        {
                printf("\nEnter the number to be entered :: ");
                scanf("%d",&data);
                rear++;
                queue[rear]=data;
                printf("\nNumber inserted is %d",queue[rear]);
                if(front==0)
                        front=1;
                return;
        }
}
void dequeue()
{
        if(front==0)
        {
                printf("\nUnderflow");
                return;
        }
        else
        {
                res=queue[front];
                if(front==rear)
                {
                        front=0;
                        rear=0;
                }
                else
                        front++;
        }
        printf("\nDeleted element is %d",res);
        return;
}
void display()
{
        int i;
        if(front==0)
        {
                printf("\nUnderflow");
                return;
        }
        for(i=front;i<=rear;i++)
                printf("%d\t",queue[i]);
}
```

2) Implement the circular queue

```
#include<stdio.h>
# define MAX 5
int cqueue[MAX];
int front = -1;
int rear = -1;
void insert(int item)
{
if((front == 0 && rear == MAX-1) || (front == rear+1))
{
printf("Queue full\n");
return;
}
if(front == -1)
{
front = 0;
rear = 0;
}
else
{
if(rear == MAX-1)
rear = 0;
else
rear = rear+1;
}
cqueue[rear] = item ;
}
void deletion()
{
if(front == -1)
{
printf("Queue Underflow\n");
return ;
}
printf("Element deleted from queue is : %d\n",cqueue[front]);
if(front == rear)
{
front = -1;
rear=-1;
}
else
{
if(front == MAX-1)
front = 0;
else
front = front+1;
}
}
void display()
{
        int front_pos = front,rear_pos = rear;
```

```
if(front == -1)
{
        printf("Queue is empty\n");
        return;
}
printf("Queue elements:");
if( front_pos <= rear_pos )
{
        while(front_pos <= rear_pos)
        {
                //printf("%d ",cqueue[front_pos]);
                //display in normal case
                printf("%d atindex[%d]\t",cqueue[front_pos],front_pos);
                front_pos++;
        }
}
else
{
        while(front_pos <= MAX-1)
        {
                //printf("%d ",cqueue[front_pos]);
        // display when rear is less than front until front reaches to max-1 and again return to 0

                printf("%d atindex[%d]\t",cqueue[front_pos],front_pos);


                front_pos++;
        }
        front_pos = 0;//return to 0 if front reaches max-1
        while(front_pos <= rear_pos)
```

```
                    {
                              //printf("%d ",cqueue[front_pos]);
                              // again display in normal case
                              printf("%d atindex[%d]\t",cqueue[front_pos],front_pos);

                              front_pos++;
                    }
          }
printf("\n");
}
int main()
{
int choice,item;
do
{
printf("\n1.Insert\n");
printf("2.Delete\n");
printf("3.Display\n");
printf("4.Quit\n");
printf("Enter your choice : ");
scanf("%d",&choice);
switch(choice)
{
case 1 :
printf("Input the element for insertion in queue : ");
scanf("%d", &item);
insert(item);
break;
case 2 :
```

```
deletion();

break;

case 3:

display();

break;

case 4:

break;

default:

printf("Wrong choice\n");

}

}while(choice!=4);

return 0;

}
```