LINEAR QUEUE:

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

#define QUE_SIZE 5

int item,front=0,rear=-1,q[10];
void insertrear()
{
   if(rear==QUE_SIZE-1)
{
            printf("queue overflow\n");
            return;
}
   printf("Enter the ITEM to be Inserted\n");
   scanf("%d",&item);

   rear=rear+1;
   q[rear]=item;

}

void deletefront()
{
   if (front>rear)
{

   front=0;
   rear=-1;
   printf("QUEUE is EMPTY\n");
```

```
}
   printf("The item deleted is %d",q[front++]);
}

void display()
{
   int i;
if (front>rear)
{
        printf("Queue is Empty\n");
}
else{
   printf("Contents of queue\n");
  for(i=front;i<=rear;i++)
  {
        printf("%d\t",q[i]);
  }
}

}

int main()
{
        int choice;
        for(;;)
        {
                printf("\n Enter the choice of Operation \n 1:Insert_Rear \n2:Delete_Front
\n3:display 4:Exit\n");
                printf("enter the choice\n");
                scanf("%d",&choice);
                switch(choice)
```

```
                {
                        case 1:insertrear ();
                        break;
                        case 2:deletefront();
                        break;
                        case 3:display();
                        break;
                        default:exit (0);

                }

        }
}
```

```
 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 1
enter the element to be inserted
654

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 1
enter the element to be inserted
6541

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 3
ITEMS of Queue
654     6541    0
 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 2
The ITEM deleted is = 654

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 2
The ITEM deleted is = 6541

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 2
Queue is EMPTY
```

CIRCULAR QUEUE :

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

```
#include<stdlib.h>

#define que_size 5
int item,front=0,rear=-1,q[que_size],count=0;
void insertrear()
{
        if(count==que_size)
        {
                printf("queue overflow\n");
                return;
        }
        printf("enter the element to be inserted\n");
        scanf("%d",&item);
        rear=(rear+1)%que_size;
        q[rear]=item;
        count++;
}
void deletefront()
{
        if(count==0) {
      printf("Queue is EMPTY\n");
        }
        else{

   item = q[front];
   printf("The ITEM deleted is = %d \n",item);
        front=(front+1)%que_size;
        count=count-1;
        }

}
```

```
void displayq()
{
	int i,f;
	if(count==0)
	{
		printf("Queue is Empty\n");
		return;
	}
	f=front;
	printf("ITEMS of Queue \n");
	for(i=0;i<=count;i++)
	{
		printf("%d\t",q[f]);
		f=(f+1)%que_size;
	}
}
int main()
{
	int choice;
	for(;;)
	{
		printf("\n Enter the choice of Operation \n1.INSERT REAR \n2.DELETE FRONT
\n3.DISPLAY  \n4.EXIT \n ");
		scanf("%d",&choice);
		switch(choice)
		{
			case 1:insertrear();
			     break;
			case 2:deletefront();
				break;
		   case 3:displayq();
```

```
                                    break;
                    default:exit(0);
                }
        }

}
```

```
 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 1
enter the element to be inserted
54810

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 1
enter the element to be inserted
06841

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 1
enter the element to be inserted
654

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 3
ITEMS of Queue
54810   6841    654     0
 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 2
The ITEM deleted is = 54810

 Enter the choice of Operation
1.INSERT REAR
2.DELETE FRONT
3.DISPLAY
4.EXIT
 2
The ITEM deleted is = 6841
```