Program 8 (A) Implement Queue using array
Theory: From notes
Code:

```
#include<stdio.h>
#define max 10
typedef struct queue
{
 int data[max];
 int front,rear;
}Q;

void init(Q *p)
{
 p->front=-1;
 p->rear=-1;
}
int is_empty(Q *p)
{
 if(p->front==-1)
 {
  printf("\n Queue is empty");
  return 1;
 }
 else
  return 0;
}
int is_full(Q *p)
{
 if(p->rear==max-1)
 {
  printf("\n Queue is full");
  return 1;
 }
 else
  return 0;
}
void InQ(Q *p,int x)
{
 if(is_full(p)==1)
  return;
 p->rear++;
 p->data[p->rear]=x;
 if(p->front==-1)
  p->front=0;
}
int DeQ(Q *p)
{
 int x;
```

```
 if(is_empty(p)==1)
  return;
 x=p->data[p->front];
 if(p->front==p->rear)
  p->front=p->rear=-1;
 else
  p->front++;
 return x;
}
void display(Q *p)
{
 if(p->front==-1)
  printf("Queue is empty");
 else
  {
   while(p->front!=max)
   {
    printf("\t%d",p->data[p->front]);
    p->front++;
   }
  }
  p->front=0;
}
void main()
{
 int x,y,z,i;
 Q Q1;
 init(&Q1);
 clrscr();
 printf("Enter elements of queue\n");
 for(i=0;i<=9;i++)
 {
  scanf("%d",&x);
  InQ(&Q1,x);
 }
 printf("\nElements of Queue are:");
 display(&Q1);
 y=DeQ(&Q1);
 printf("\nDeleted element is %d\n",y);
 z=DeQ(&Q1);
 printf("\nDeleted element is %d",z);
 getch();
}
```

OUTPUT:

```
Enter elements of queue
10
20
30
40
50
60
70
80
90
100

Elements of Queue are:  10      20      30      40      50      60      70
80      90      100
Deleted element is 10

Deleted element is 20_
```