AIM: Implement Circular Linear Queue ADT using array.

CODE:

#include<stdio.h>

#include<conio.h>

#define SIZE 10

int queue[SIZE];

int R=-1;

int F=-1;

int i;

void enqueue(int value)

{

if((F==0 && R==SIZE-1) || F==R+1)

{

printf("Queue is full\n");

}

else

{

if(F==-1)

{

F=0;

}

R=(R+1)%SIZE;

queue[R]=value;

}

}

void dequeue()

{

if(F==-1)

{

printf("Queue is empty\n");

}

else

{

printf("Deleted element: %d", queue[F]);

if(F==R)

{

F=-1;

R=-1;

}

else

{

F=(F+1)%SIZE;

}

}

}

void peek()

{

if(F==-1)

{

printf("Queue is empty\n");

}

else

{

printf("The front element is: %d\n", queue[F]);

}

}

void display()

{

if(F==-1)

{

printf("Queue is empty\n");

}

else

{

printf("The Queue is: \n");

for(i=F; i<=R; i++)

{

printf("%d\t", queue[i]);

}

}

}

void main()

{

int choice;

int x;

clrscr();

printf("---Circular Queue implementation---\n\n");

while(1)

{

printf("\n\nOperations performed by Queue");

printf("\n1. Enqueue\n2. Dequue\n3. Peek\n4. Display\n5. End");

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

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("\nEnter the element to be inserted: ");

scanf("%d", &x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

peek();

break;

case 4:

display();

break;

case 5:

exit(0);

break;

default:

printf("Invalid choice\n");

}

}

getch();

}

OUTPUT: