Aim:-Circular Queue using Array

Algorithm:-

ENQUEUE(DATA)

Create a struct node type node.

Insert the given data in the new node data section and NULL in address section.

If Queue is empty then initialize front and rear from new node.

Queue is not empty then initialize rear next and rear from new node.

New node next initialize from front

DEQUEUE()

Check if queue is empty or not.

If queue is empty then dequeue is not possible.

Else Initialize temp from front.

If front is equal to the rear then initialize front and rear from null.

Print data of temp and free temp memory.

If there is more than one node in Queue then make front next to front then

initialize rear next from front.

Print temp and free temp.

PRINT()

Check if there is some data in the queue or not.

If the queue is empty print “No data in the queue.”

Else define a node pointer and initialize it with front.

Print data of node pointer until the next of node pointer becomes NULL.\*

Program:-

#include <stdio.h>

# define max 6

int queue[max]; // array declaration

int front=-1;

int rear=-1;

// function to insert an element in a circular queue

void enqueue(int element)

{

if(front==-1 && rear==-1) // condition to check queue is empty

{

front=0;

rear=0;

queue[rear]=element;

}

else if((rear+1)%max==front) // condition to check queue is full

{

printf("Queue is overflow..");

}

else

{

rear=(rear+1)%max; // rear is incremented

queue[rear]=element; // assigning a value to the queue at the rear position.

}

}

// function to delete the element from the queue

int dequeue()

{

if((front==-1) && (rear==-1)) // condition to check queue is empty

{

printf("\nQueue is underflow..");

}

else if(front==rear)

{

printf("\nThe dequeued element is %d", queue[front]);

front=-1;

rear=-1;

}

else

{

printf("\nThe dequeued element is %d", queue[front]);

front=(front+1)%max;

}

}

// function to display the elements of a queue

void display()

{

int i=front;

if(front==-1 && rear==-1)

{

printf("\n Queue is empty..");

}

else

{

printf("\nElements in a Queue are :");

while(i<=rear)

{

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

i=(i+1)%max;

}

}

}

int main()

{

int choice=1,x; // variables declaration

while(choice<4 && choice!=0) // while loop

{

printf("\n Press 1: Insert an element");

printf("\nPress 2: Delete an element");

printf("\nPress 3: Display the element");

printf("\nEnter your choice");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter the element which is to be inserted");

scanf("%d", &x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

}}

return 0;

