**Date:** 10/01/2023

**Program 8(a):** Implementation of circular queue using array.

**Algorithm:**

Step 1: Initialize an array queue of size n, where n is the maximum number of elements that the queue can hold.

Step 2: initialize two variables front and rear to -1.

Step 3: To enqueue an element x onto the queue, do the following:

* Increment by 1
* if rear is equal to n, set rear to 0.
* if front is -1, set front to 0
* set queue[rear] to x

Step 4: To dequeue an element from the queue, do the following:

* check if the queue is empty by checking if front is -1. If it is, return an error message indicating that the queue is empty.
* set x to queue[front]
* if front is equal to rear, set front and rear to -1
* otherwise, increment front by 1 and if front is equal to n, set front to 0
* return x

**Program code:**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 5

int items[SIZE];

int front = -1, rear = -1;

// Check if the queue is full

int isFull() {

if ((front == rear + 1) || (front == 0 && rear == SIZE - 1))

return 1;

return 0;

}

// Check if the queue is empty

int isEmpty() {

if (front == -1)

return 1;

return 0;

}

// Adding an element

void enQueue(int element) {

if (isFull())

printf("\n Queue is full\n");

else

{

if (front == -1) front = 0;

rear = (rear + 1) % SIZE;

items[rear] = element;

printf("\n Inserted %d", element);

}

}

// Removing an element

int deQueue() {

int element;

if (isEmpty()) {

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

return (-1);

}

else

{

element = items[front];

if (front == rear)

{

front = -1;

rear = -1;

}

// Q has only one element, so we reset the

// queue after dequeing it. ?

else

{

front = (front + 1) % SIZE;

}

printf("\n Deleted element %d \n", element);

return (element);

}

}

// Display the queue

void display() {

int i;

if (isEmpty())

printf(" \n Empty Queue\n");

else

{

printf("\n Front %d ", front);

printf("\n Items ");

for (i = front; i != rear; i = (i + 1) % SIZE)

{

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

}

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

printf("\n Rear %d \n", rear);

}

}

int main()

{

int choice;

int x;

while(1)

{

printf("\nOperations to be performed are\n");

printf("1. Insert an element in the queue\n");

printf("2. Delete an element in the queue\n");

printf("3. Display the elements in the queue\n");

printf("4. Exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1: printf("Enter the element u want to insert:");

scanf("%d",&x);

enQueue(x);

break;

case 2: deQueue();

break;

case 3: display();

break;

case 4: exit(0);

break;

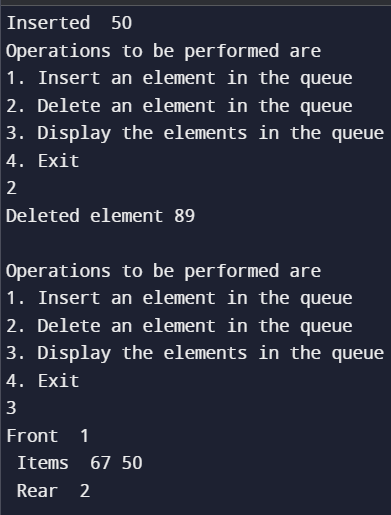
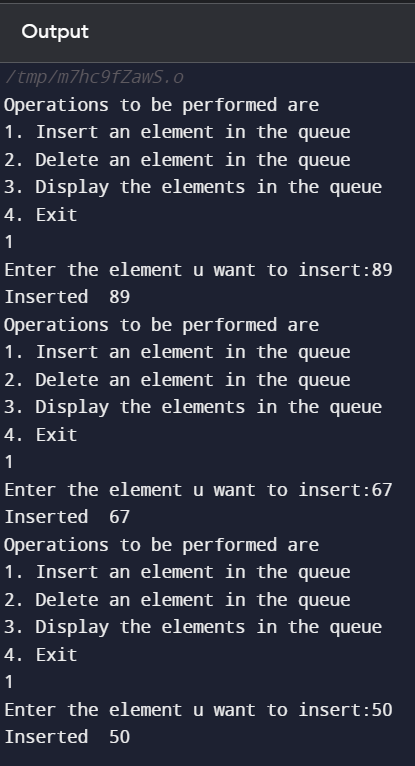
default:printf("Invalid choice");

}

}

}

**Output:**

****

**Link for GitHub:**