Program for implementation of circular queue using array

ALGORITHM

A)Enqueue(x) Operation

You should follow the following steps to insert (enqueue) a data element into a circular queue -

Step 1: Check if the queue is full (Rear + 1 % Maxsize = Front)

Step 2: If the queue is full, there will be an Overflow error

Step 3: Check if the queue is empty, and set both Front and Rear to 0

Step 4: If Rear = Maxsize - 1 & Front != 0 (rear pointer is at the end of the queue and front is not at 0th index), then set Rear = 0

Step 5: Otherwise, set Rear = (Rear + 1) % Maxsize

Step 6: Insert the element into the queue (Queue[Rear] = x)

Step 7: Exit

B) Dequeue() Operation

Step 1: Check if the queue is empty (Front = -1 & Rear = -1)

Step 2: If the queue is empty, Underflow error

Step 3: Set Element = Queue[Front]

Step 4: If there is only one element in a queue, set both Front and Rear to -1 (IF Front = Rear, set Front = Rear = -1)

Step 5: And if Front = Maxsize -1 set Front = 0

Step 6: Otherwise, set Front = Front + 1

Step 7: Exit

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;

}

OUTPUT:

