12/10/2020

Lab 4 - Queue Implementation.

WAP to simulate the working of a queue of integers using an array. Prove the following operations. a) Insert b) Delete c) Display. The program should print appropriate messages for queue empty and queue overflow conditions.

Pseudocode:

```
Queue[MAX]
Front ← -1
rear ← -1

Insert()
if rear = MAX-1
   Printf("Overflow")
else if front = = -1 and rear = -1
        front = rear = 0
else
        rear = rear+1
        Queue[rear] = num

delete()
if front = -1 or front > rear
    Print("Underflow)
else
      val = Queue[front]
      front = front+1
```

```
display()
 if (front = -1 || front > rear)
   print (Queue is Empty)
else
     print queue //
```

C program:

```
#include <stdio.h>
#define MAX 10
int queue[MAX];
int front = -1 , rear = -1;
void insert();
Int delete_element();
void display();
int main()
{
int option, val;
do
{
 printf("\n ------");
 printf("\n Options:");
 printf("\n 1. Insert an element");
 printf("\n 2. Delete an element");
 printf("\n 3. Display the queue.");
 printf("\n 4. Exit");
 printf("\n -------");
 printf("\n Enter your option:");
 scanf("%d", &option);
```

```
    switch(option)
    {
case 1:
    insert();
    break;
case 2:
val = delete_element();
if(val != -1)
printf("\n The number deleted is : %d", val);
break;
case 3:
    display();
    break;
}
}
while(option!=4);
return 0;
}
void insert()
{
    int num;
    printf("\n Enter the number to be inserted in
                the queue:");
    scanf("%d",&num);
    if (rear = MAX-1)
    printf("\n Overflow has occured in the
                queue");
    else if(front == -1 && rear == -1)
    front = rear = 0;
```

```
else
rear++;
queue[rear]=num;
}
int delete_element()
{
  int val;
if (front == -1 || front > rear)
{
  printf("\n Underflow has occured in the queue
  return -1;
}
else
{
 val=queue[front];
 front++;
 if (front>rear)
 front=rear=-1;
  return val;
}
}
void display()
{
  int i;
  printf("\n");
 if (front==-1 || front>rear)
 printf("\n Queue is Empty");
 else
  for (i=front; i<=rear; i++)
  printf("\t %d", queue[i]);
}
```