lab - 4

Date: 08 01 2024

Queue

1. Write a program to simulate the working of the queue of integers using an array. Provide the following operations: Insert, delete, display. The program should print appropriate message for overflow and underflow condition.

```
void insert()
{
 int num;
 printf("\n Enter the number to be inserted into queue: ");
 scanf("%d", &num);
 if (rear == MAX - 1)
 {
  printf("\n Overflow ");
 else if (front == -1 && rear == -1)
  {
   front = 0;
   rear  = 0;
  }
  else
   rear++;
 q[rear] = num;
}
```

```
void delete ()
{
 int val;
 if (front == -1 || front > rear)
 {
  printf ("In underflow");
  return -1;
 }
 else
 {
 val = q[front];
 front++;
 if (front > rear)
 {
  rear = -1;
  front = -1;
 }.
return val;
 }
}.
```

```
void display()
{
 if (front == -1)
   printf ("Empty Queue\n");
 else
  {
   printf ("Queue: \n");
   for (int i = front; i <= rear; i++)
   printf ("%d", q[i]);
  }
}
```

Output:

```
---Menu---
enter 1: to ingert
enter 2: to delete
enter 3: to display
enter 4: exit.
```

```
Enter your choice = 1
Enter the number to ingert: 62
Enter your choice: 2
Deleted element is: 62
Enter your choice: 3.
Enterment of queue: 42
element of queue: 27
Enter your choice: 4
exit.
```

2. Circular Queue

```
int isFull()
{
  if ((front == rear+1) || (front == 0 && rear == size-1))
  return 1;
  return 0;
}
int isEmpty() {
 if (front == -1)
 return 1;
 return 0;
}

void enqueue (int element)
{
 if (isFull())
 printf ("\n Queue is Full");
 else
 if (front == -1)
 front=0;
 rear = (rear+1) % size;
 items[rear] = element;
 printf ("\n Inserted -> %d", element);
}
}
```

```
int dequeue() {
int element;
if (isEmpty()) {
  printf ("Queue is Empty");
  return (-1);
}
else {
  element = items[front];
  if (front == rear)
  {
   front = -1;
   rear  = -1;
  }
  else {
  front = (front+1)%size;
}
printf ("\n Deleted element -> %d ", element);
return (element);
}
}
```

```
void display() {
  int i;
  if (isEmpty())
  printf (" Empty Queue ");
  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 ", rear);
  }
}
```

8w 8/11/24

Output: *** Menu ***

1. Ingcrut
2. Delete
3. Dipplay
4. exit

Enteer your choice: 1
enter the element: 5
Enter your choice: 2
deleted element 1.

Enter your choice: 3
underflow.
Enter your choice: 4
exit.