```
LAB-3
4. LINEAR QUEUE USING ARRAYS
# include estolio.b>
# include co
# define MAX 16
int quene[NAX], front=+, rear=+;
rold insert () f
  gut num:
   Prints ("Enter number to be "inserted: ");
  scend ("xd", 7 num);
  ? ( rear = = MAX -1) {
    prints (" in overstows!");
   else if (front =- 1) {
     front = rear=0;
   else {
     rear +1;
      queue [rear] = num;
 3 <
 int delete () {
   ; to tu?
    If (front == 1 || front > rear) {
       print ("In Underflow!");
       return -1;
    else {
       ral = queue [front];
       front ++;
       if (front > rear) {
          front = rear = +;
       return val;
 int peek () {
   if (front == -1 || front > rear) {
       printf ("In Quese Empty!");
        return -1;
   else {
       return queue [front];
```

```
rold displaye) {
  of (front == -1 || front > rear) {
    prints ("In Queue empty!");
  else {
    for Li= front; " = rear; "++) {
       printf E"In Vd", quene(i));
grt main () {
  int choice, value;
  printf ("In Queve Oberation Menu");
   printf ("In 1. Ansert In 2. Delete In 3. Peels In 4. Display In 5. Exit");
   printef ("In Enter your choice: ");
   scanf ("Tid", 4 aphroe);
   switch (choice) {
     casel: "insert();
             break;
     case 2: ralv= delete ();
             if (ral] = -1) {
               prints (" In Defeted number: "d", value);
             break;
     case 3: value = peck (1;
             if (val) = -1) {
               print ("In First value in quene: "Id", value);
             break;
     case 4: display ();
             break;
     Case 5: exit (0);
     default: printf ("Apput Invalid!");
   return 01
```

```
main.c
                                                                                                         Oueue Operations:
                                                                                                         1. Insert 2. Delete 3. Display 4. Exit
 1 #include <stdio.h>
                                                                                                         Enter your choice: 2
 2 #include <stdlib.h>
                                                                                                         Queue Underflow: Cannot delete element, queue is empty
 4 #define MAX_SIZE 3
                                                                                                         Queue Operations:
                                                                                                         1. Insert 2. Delete 3. Display 4. Exit
 6 int queue[MAX_SIZE];
 7 int front = -1, rear = -1;
                                                                                                         Enter your choice: 1
                                                                                                         Enter the value to insert: 4
9 - void insert(int value) {
                                                                                                         Element 4 inserted into the queue
       if (rear == MAX_SIZE - 1) {
          printf("Queue Overflow: Cannot insert element %d, queue is full\n", value);
11
                                                                                                        Queue Operations:
       } else {
12 +
                                                                                                         1. Insert 2. Delete 3. Display 4. Exit
          if (front == -1) {
13 +
                                                                                                         Enter your choice: 1
              front = 0;
14
                                                                                                         Enter the value to insert: 5
15
                                                                                                         Element 5 inserted into the queue
           rear++;
16
          queuefrear1 = value;
17
           printf("Element %d inserted into the queue\n", value);
                                                                                                         Queue Operations:
18
                                                                                                         1. Insert 2. Delete 3. Display 4. Exit
19
                                                                                                         Enter your choice: 1
20 }
                                                                                                         Enter the value to insert: 6
21
                                                                                                         Element 6 inserted into the queue
22 - void delete() {
       1f (front == -1) {
          printf("Queue Underflow: Cannot delete element, queue is empty\n");
                                                                                                         Queue Operations:
25 *
       } else {
                                                                                                         1. Insert 2. Delete 3. Display 4. Exit
           printf("Element %d deleted from the queue\n", queue[front]);
26
                                                                                                         Enter your choice: 1
          if (front == rear) {
27 +
                                                                                                         Enter the value to insert: 7
              // Reset gueue when the last element is deleted
28
                                                                                                         Queue Overflow: Cannot insert element 7, queue is full
29
              front = rear = -1;
          } else {
30 +
                                                                                                        Queue Operations:
31
              front++;
                                                                                                         1. Insert 2. Delete 3. Display 4. Exit
32
                                                                                                         Enter your choice: 2
33
                                                                                                      - Element 4 deleted from the queue
20 2
```

```
2. CIRCULAR QUEUE USING ARRAYS
  # Enolude xetalio. b
  # define MAX B
  int queue [MAX];
  Put front = -1, repr = -1)
  void Ensert()[
    printf ("In Exter number to be inserted: ");
    Int num;
    scenf ("Y.d", fnum);
    of (fort == 0 of rear = MAX-1) {
      printf ["In acoflow");
   else if (front == -1 4 + rear == -1) {
      front = rear=0)
      queue [rear] = num;
   else if (rear=-MAX-1 &f front!=0) {
      rear=0;
      queue [rear]=num;
   else ?
      rearty)
     greve [reer]=num)
and delete () {
  int val;
  if ( front == + && rear == 1) {
    printf("In underflow!);
    return -1;
  rd = queue [from];
  if (front = rear) &
    front = recr = -1;
    if (front == MAX-1) {
    else £
      front #;
return vol;
```

```
word display() ?
  (int i)
   printefront == 1 At rear == -1)[
     printf ("In Queue Empty");
     lif Cfront a rear E
        for ( i= front ) i erear; i+1) {
          printf ("In t.d", quene [i]);
  Helse &
        for ( i=front; i < MAX; i+) {
          printf ("In r.d", quene (i));
        for (i=0; ic=rear) i+1) {
          printf("In 'd') quand (i));
   星子
Put main () &
  int choice, value,
  printf (" Queus operation Menu");
  printf ("In1, Insert In 2. Delete In 3. Display to 4. Exit);
  printf ("Enter a choice: ");
  scent ("Y.d", Echoice);
  switch (choice) ?
    casel: insert();
        break;
    case 2: notedeletel;
                            o if (rold=1) printy ("Deletet-number: Y.d", rehal)
            break;
    case 3: display ()
           breck;
    Case 4: exit (6);
    default: printf ("Apput Armolid!");
 a return of
```

```
0
                                                                                                              Output
main.c
                                                                                          Save
                                                                                                    Run
                                                                                                             CAT CHANT QUEUE OPET HEADTH.
    #include <stdio.h>
                                                                                                             1. Insert 2. Delete 3. Display 4. Exit
    #include <stdlib.h>
                                                                                                             Enter your choice: 1
                                                                                                             Enter the value to insert: 4
    #define MAX_SIZE 3
                                                                                                             Element 4 inserted into the circular queue
 6 int circularOueue(MAX SIZE1:
                                                                                                             Circular Oueue Operations:
    int front = -1, rear = -1:
                                                                                                             1. Insert 2. Delete 3. Display 4. Exit
                                                                                                             Enter your choice: 1
 9 - void insert(int value) (
                                                                                                             Enter the value to insert: 5
        if ((rear + 1) % MAX SIZE == front) {
10 -
                                                                                                             Element 5 inserted into the circular queue
            printf("Queue Overflow: Cannot insert element %d, circular queue is full\n", value):
12 +
       } else {
                                                                                                             Circular Queue Operations:
            if (front == -1) {
13 +
                                                                                                            1. Insert 2. Delete 3. Display 4. Exit
14
               front = 0:
                                                                                                             Enter your choice: 1
15
                                                                                                             Enter the value to insert: 6
            rear = (rear + 1) % MAX_SIZE;
16
                                                                                                             Element 6 inserted into the circular queue
            circularQueue[rear] = value;
17
            printf("Element %d inserted into the circular gueue\n", value);
18
                                                                                                             Circular Queue Operations:
19
                                                                                                             1. Insert 2. Delete 3. Display 4. Exit
20 }
                                                                                                             Enter your choice: 1
21
                                                                                                             Enter the value to insert: 7
22 - void delete() {
                                                                                                             Queue Overflow: Cannot insert element 7, circular queue is full
        if (front == -1) {
            printf("Queue Empty: Cannot delete element, circular queue is empty\n");
24
                                                                                                             Circular Queue Operations:
25 +
       ) else (
                                                                                                             1. Insert 2. Delete 3. Display 4. Exit
            printf("Element %d deleted from the circular queue\n", circularQueue[front]);
26
                                                                                                             Enter your choice: 2
27 -
           if (front == rear) {
                                                                                                             Element 4 deleted from the circular queue
                front = rear = -1;
28
29 +
           ) else (
                                                                                                             Circular Queue Operations:
                front = (front + 1) % MAX SIZE:
30
                                                                                                             1. Insert 2. Delete 3. Display 4. Exit
31
                                                                                                             Enter your choice: 1
32
                                                                                                             Enter the value to insert: 8
33 )
                                                                                                          * Element 8 inserted into the circular queue
20.00
```

```
2. Output!
```

Circular Quene operation Menu

- 1. Ansest
- 2. Delete
- 3. Display
- q. exit

Enter your choice! 2

Underfton.

Enter your choice: 1

Entor a number: 20

Enter your choice: 1

Enter a number: 40

enter your choice: 1

Exter a number: 60

Enter your choice: 1

Enter a number: 80

Overflow.

enter your choice : 3

20

40

60

Enter your choice: 4

exit

## Output:

Liher Quene operation Meny

- 1. Ansest
- 2. Delete
- 3. Display
- 4. exit

enter your choice (2

underflow.

Enter your choice: 1

Enter a number: 20

Enter your choice, 1

enter a number: 40

Enter your choice.)

Enter a number: 60

Sept.

Enter your choice: 1
Enter a number: 80
Overflow.

Enter your choice 3

20

40

60