

ILAHIA COLLEGE OF ENGINEERING
AND TECHNOLOGY

DEPARTMENT OF MCA.

FIRST SEMESTER MCA (2020 SCHEME) PRACTICAL
EXAMINATION JUNE 2021

20MCA135 DATA STRUCTURE LAB

Date: 30-6-2021

Time: 9:30am - 12:30am

Submitted by,

Aloshya Joshy.

ICE20MCA2008

Batch: B

- 1) Merging of two arrays?
- 2) Implement circular queue?

```
2) #include <stdio.h>
#include <conio.h>
#define MAX 5
```

```
int cqueue-arr[MAX];
```

```
int front = -1;
```

```
int rear = -1;
```

```
void insert (int item)
{
```

```

if ((front == 0 && rear == MAX-1) || (front == rear+1))
{
    printf("Queue underflow\n");
    return 0;
}

```

```

if (front == -1)

```

```

{
    front = 0;
    rear = 0;
}

```

```

else
{

```

```

    rear = rear + 1;
    rear = 0;

```

```

if (rear == MAX-1)

```

```

    rear = 0;

```

```

else

```

```

    rear = rear + 1;

```

```

}

```

```

cqueue_arr[rear] = item;

```

```

}

```

```

void deletion()

```

```

{

```

```

if (front == -1)

```

```

{

```

```

    printf

```

```

    printf("Queue underflow\n");

```

```

    return;

```

```

}

```

```

printf("Elements deleted from queue is: %d\n",
    cqueue_arr[front]);

```

```
if (rear == front)
```

```
if (front == rear)
```

```
{
```

```
front = -1;
```

```
rear = -1;
```

```
}
```

```
else
```

```
{
```

```
if (front == MAX-1)
```

```
front = 0;
```

```
else
```

```
front = front + 1;
```

```
}
```

```
}
```

```
void display()
```

```
{
```

```
int front_pos = front, rear_pos = rear;
```

```
if (front == -1)
```

```
if (front == -1)
```

```
{
```

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

```
return;
```

```
}
```

```
printf("Queue elements : ");
```

```
scanf
```

```
if (front_pos <= rear_pos)
```

```
while (front_pos <= rear_pos)
```

```
{
```

```
printf("%d", queue-arr[front_pos]);
```

```
front_pos++;
```

```
}
```

```
else
```

```
{
```

```
while (front - pos <= MAX-1) while (front - pos <= MAX-1)
{
    printf("%d", queue-arr[front-pos]);
    front-pos++;
}
```

```
front-pos++;
```

```
front-pos = 0;
```

```
while (front - pos <= rear-pos)
```

```
{
    printf("%d", queue-arr[front-pos]);
    front-pos++;
}
```

```
void main()
```

```
{
    int choice, item;
```

```
clrscr();
```

```
printf("\n");
```

```
}
```

```
void main()
```

```
{
```

```
int choice, item;
```

```
clrscr();
```

```
do
```

```
{
```

```
printf("Insert the element for insertion in queue.");
```

```
scanf("%d", &item);
```

```
printf("\n Insert Circular queue \n");
```

```
printf("1. Insert \n");
```

```
printf("2. Delete \n");
```

```
printf("3. Display \n");
```

```
printf("4. Quit \n");
```



```

switch(choice)
printf("Enter your choice : ");
scanf("%d", &choice);
switch(choice)
{
case 1:
    printf("Insert the element for insertion in queue : ");
    scanf("%d", &item);
    insert(item);
    break;
case 2:
    deletion();
    break;
case 3:
    display();
    break;
case 4:
    exit();
default:
    printf("Wrong choice \n");
}
while(choice != 4);
getch();
}

```

Output

circular queue

1. Insertion
2. Delete
3. Display.
4. Quit

Enter your choice : .

Insert the element for insertion in queue: 12.

Circular Queue

1. Insert
2. Delete
3. Display.
4. Quit.

Enter your choice : 3

Queue elements : 12.

Circular Queue

1. Insert
2. Delete
3. Display
4. Quit.

Enter your choice : 2

Element deleted from queue is 12.

Circular Queue.

1. Insert
2. Delete
3. Display.
4. Quit.

Enter your choice : 4

Algorithm

Step 1: start

Step 2: write function for insert an element; delete the inserted elements and display both inserted and deleted elements. An insertion without happened in the over the

underflow state.

step 3: using switch statement to call the insertion, deletion and display functions.

step 4: stop.

1) Merging of two arrays?

Algorithm

step 1: start

step 2: initialize variables

step 3: access the size of the array and insert sorted elements.

step 4: check array 1 is less than array 2. then insert the values of array 1 to array 3.
else,

insert the value of array 2 to array 3.

step 5: then perform the merge operation

step 6: stop.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main()
```

```
int array1[25], array2[25], array3[50], m, i, j, n, k=0;  
clrscr();
```

```
printf("enter the size of the array array 1:");  
scanf("%d", &m);
```

```
printf("Enter sorted elements of array 1:");
```

```
scanf("%d", &n);
```

```
for (i=0; i<n; i++)
```

```
{
```

```
scanf("%d", &array1[i]);
```

```
}
```

```
printf("\n Enter the size of the array 2:");
```

```
scanf("%d", &n);
```

```
printf("\n Enter sorted elements of array 2:");
```

```
for (i=0; i<n; i++)
```

```
{
```

```
scanf("%d", &array2[i]);
```

```
}
```

```
i=0;
```

```
j=0;
```

```
while (i<m && j<n)
```

```
{
```

```
if (array1[i] < array2[j])
```

```
{
```

```
array3[k] = array1[i];
```

```
i++;
```

```
}
```

```
else
```

```
{
```

```
array3[k] = array2[j];
```

```
j++;
```

```
}
```

```
k++;
```

```
}
```

```
if (i >= m)
```

```
{
```

```
while (j < n) while (i < n)
```



```

{
array 3[k] = array 2[j];
array 3[k] = array 1[i]; array 3[k] = array 2[j];
    j++;
    k++;
}
}
if (j >= n)
{
while (i < m)
{
array 3[k] = array 2[i]; array 3[k] = array 1[i];
    i++;
    k++;
printf("\n After Merging:\n");
for (i = 0; i < m+n; i++)
{
printf("\n %d", array 3[i]);
}
getch();
}

```

Output

Enter the size of 1st array : 3

Enter the sorted elements of 1st array :

1 2 3

Enter the size of 2nd array : 3

Enter the elements of 2nd array :

5 6 7

After Merging:

1

2

3

5

6

7