FIRST SEMESTER MCA (2020 SCHEME) PRACTICAL EXAMINATION JUNE-JULY 2021 20MCA135 DATA STRUCTURES LAB	ANAGHA M KUMAR ICE 20MCA-2009
Date - 30.06.2021 Batch - II	Time - 9.30-12.30

Program 1 - Merging of two arrays.

Algorithm

Step 1 - Start

Step 2 - Declare 3 arrays, array 1, array 2 and array 3.

Step 2 - Read the values of arrays.

Step 3 - Compare array 1 and array 2 using while.

Step 4 - Merge the elements of two Sorted array.

Step 5 - Display the array after Merging.

Step 6 - Stop.

Default input and output

Enter the size of first array : 2

Enter the sorted elements of first array

1

2

Enter the size of second array: 3

Enter the sorted elements of second array:

3

7

8

after merging :

1

2

3

7

8

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int array1[50], array2[50], array3[100], m,n,
          i, j, k=0;
  clrscr();
  Printf("\n Enter the size of first array :");
  Scanf("%d", &m);
Printf("\n Enter the sorted elements of first
                                    array \n");
```

```
for (i=0; i<m; i++)
{
    scanf ("%d", &array1[i]);
}
printf ("\n Enter the size of Second array ");
 scanf ("%d", &n);
 printf ("\n Enter Sorted elements of Second
        array \n");
 for (i=0; i<n; i++)
 {
   scanf ("%d", & array2[i]);
 }
  i=0;
 j= 0;
 while (i<m && j<n)
 {
 if (array1[i] < array2[j])
 {
 array 3[k]= array1[i];
 i++;
 }
else
{
  array3[k]= array2[j];
 j++;
```

```
}
k++;
}
if (i>=m)
{
 while (j<n)
 {
  array3[k]= array2[j];
  j++;
  k++;
 }
}
 if (j>=n)
 {
 while (i<m)
 {
  array3[k]= array1[i];
  i++;
  k++;
  }
Printf ("\n  after merging : \n");
  for (i=0 ; i<m+n; i++)
  {
   Printf ("%d ;  array3[i]);
  }
  getch(); }
```

Program 2 - Implement Circular Queue.

Algorithm

Step 1 - Start

Step 2 - Declare Circular Queue and front and rear.

Step 2 - check the Queue is overflow. then give the value of front and rear.

Step 3 - 3 function are ~~declared~~, here, that is deletion ~~add~~, display. and main.

Step 4 - In the deletion function performed element deleted from Queue.

Step 5 - And the display function performed check the Queue is empty and display the Queue elements.

Step 6 - The main function peerforme the operations Insert, delete, display, Quit .... In the Switch Statement using. ~~And de~~

Step 7 - Stop.

Output

Enter the Size

Default input and output

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

Enter your choice : 1

Insert the element for insertion in Queue: 10

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

Enter your choice : 1

Insert the element for insertion in Queue - 12

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

Enter your choice - 3

Queue elements : 10 12

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

Enter your choice : 2

Element deleted from Queue is : 10

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

Enter your choice : 4

Program

```
#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 overflow\n");
        return;
    }
    if (front == -1)
    {
        front = 0
        rear = 0;
    }
    else
    {
        if (rear == MAX-1)
            rear = 0;
```

```
}
else
{
    if (rear == MAX-1)
            rear = 0;
        else
            rear = rear + 1;
    }
    cqueue_arr[ rear ] = item;
}
void deletion()
{
    if (front == -1)
    {
        printf ("Queue is underflow\n");
        return;
    }
    printf ("Element deleted from queue is:
                %d\n", cqueue_arr[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)
        {
            printf("Queue is Empty\n");
            return;
        }
        printf("Queue elements :");

    front_pos = 0;
    while (front_pos <= rear_pos)
    {
        printf("%d", queue_arr[front_pos]);
        front_pos++;
    }
    printf("\n");
}
void main()
{
    int choice, item;
    do
    {
    printf("1. Insert\n");
    printf("2. Delete\n");
    printf("3. Display\n");
    printf("4. Quit\n");
    printf("Enter your choice : ");
    scanf("%d", &choice);
    switch (choice)
    {
```

```
case 1:
 printf("Input the element for insertion
                    in Queue : ");
    scanf("%d", &item);
     insert(item);
      break;
     case 2:
      deletion();
       break;
      case 3:
       display();
        break;

       case 4:
        break;
       default;
        printf("Wrong choice \n");
      }
   }
  while(choice != 4);
   getch();
}
```