Assignment Day 7 | 31st December 2020

Question-1:

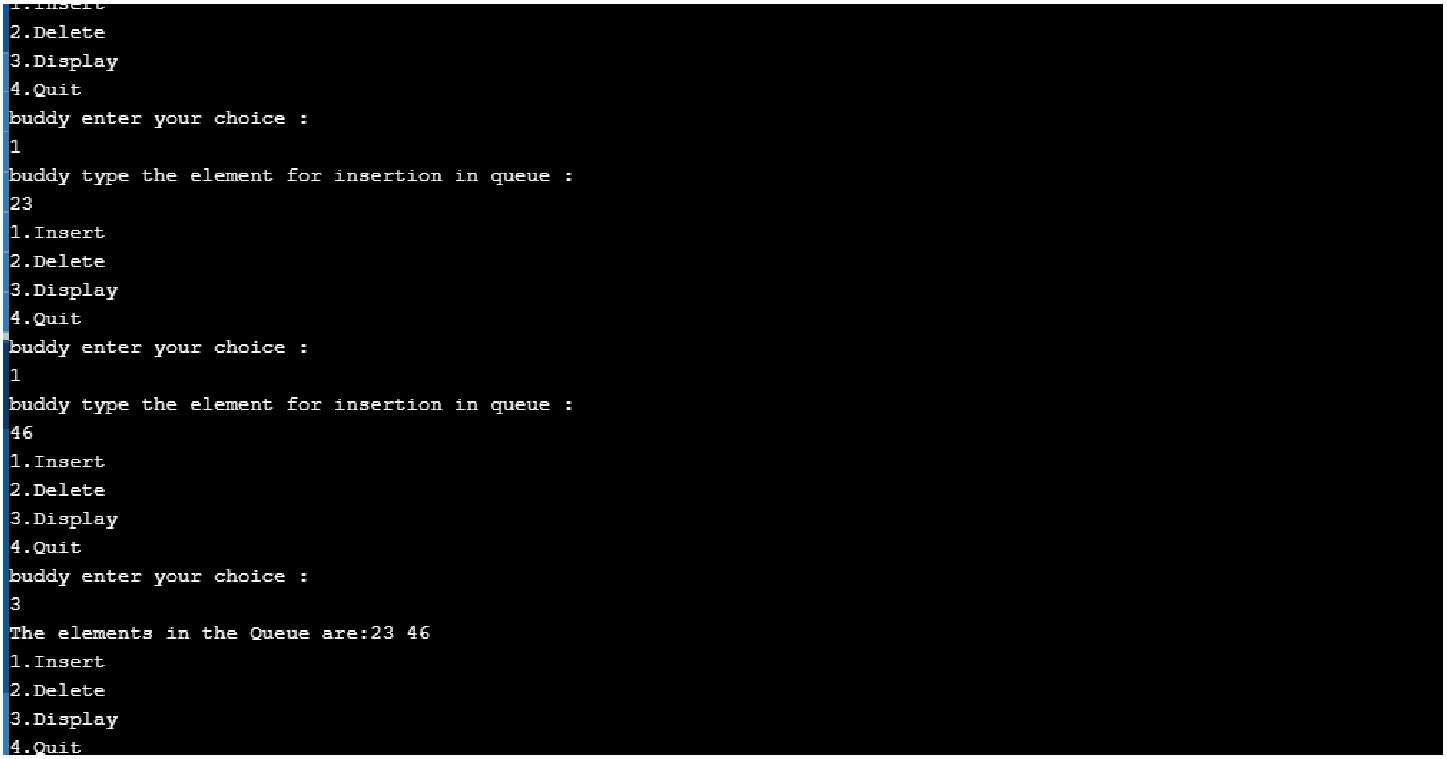
Write a program implementing insert, delete and display operation of Circular Queue.

C-Code:

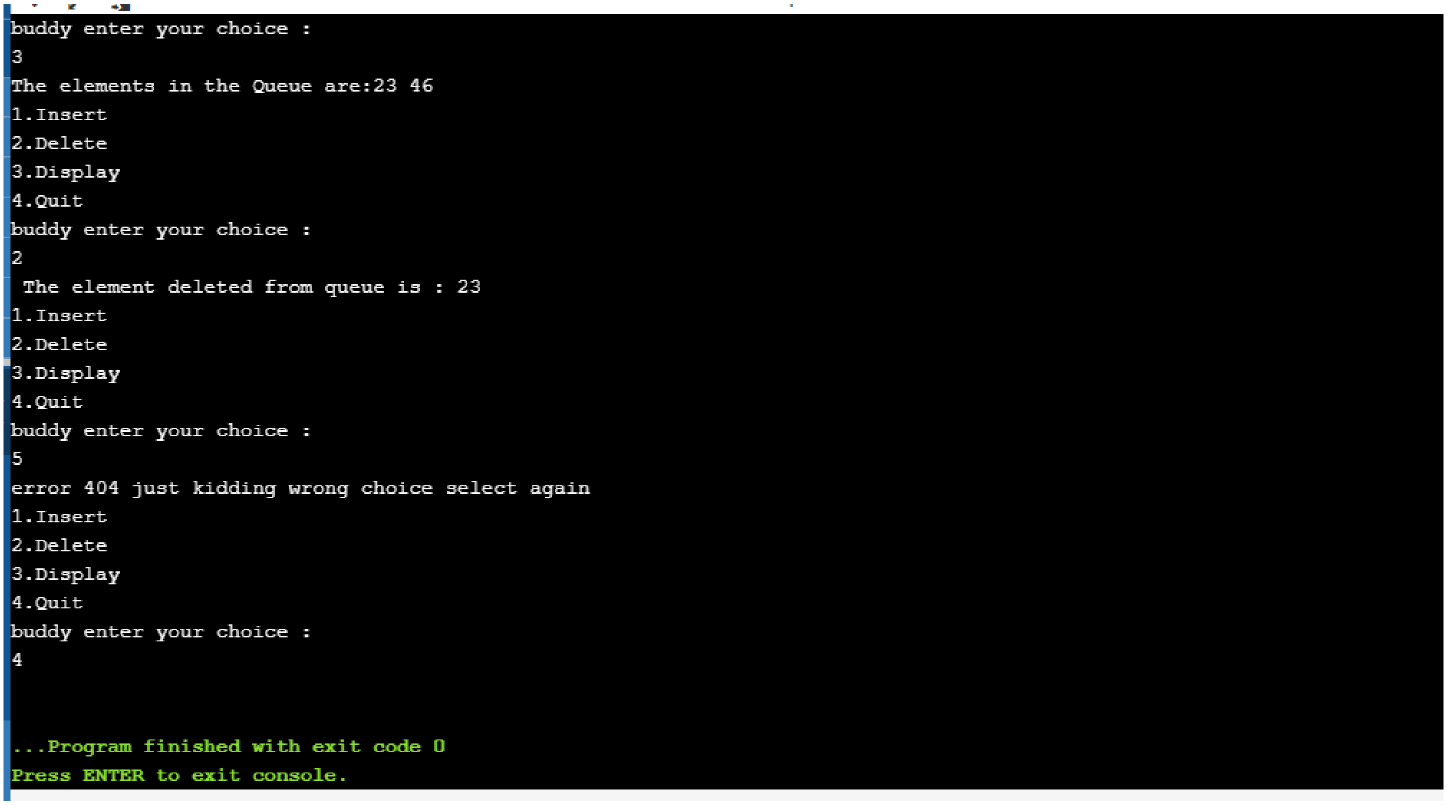
```
#include<stdio.h>
#define MAX 5
int Circular_Queue_array[MAX];
int front = -1, rear = -1;
void Insert(int items)
{
        if ((front == 0 && rear == MAX - 1) || (front == rear + 1))
        {
                printf("buddy the Queue overflowing \n");
                return;
        }
        if (front == -1)
        {
                front = 0;
                rear = 0;
        }
        else
        {
                if (rear == MAX - 1)
                rear = 0;
                else
                rear = rear + 1;
        }
        Circular_Queue_array[rear] = items;
```

```
}
void Deletion()
{
    if (front == -1)
    {
        printf("buddy the Queue underflowing (i.e, its empty)\n");
        return;
    }
    printf(" The element deleted from queue is : %d \n", Circular_Queue_array[front]);
    if (front == rear)
    {
        front = -1;
        rear = -1;
    }
    else
    {
        if (front == MAX - 1)
        front = 0;
        else
        front = front + 1;
    }
}
void Display()
{
    int front_position = front, rear_position = rear;
    if (front == -1)
    {
        printf("buddy the Queue is empty, add some the access\n");
        return;
    }
    printf(“The elements in the Queue are:");
    if (front_position <= rear_position)
    while (front_position <= rear_position)
    {
        printf("%d ", Circular_Queue_array[front_position]);
        front_position++;
    }
    else
    {
        while (front_position <= MAX - 1)
        {
            printf("%d ", Circular_Queue_array[front_position]);
            front_position++;
        }
            front_position = 0;
            while (front_position <= rear_position)
            {
                printf("%d ", Circular_Queue_array[front_position]);
                front_position++;
            }
    }
    printf("\n");
}
int main()
{
    int choice, items;
    do
    {
        printf("1.Insert\n");
        printf("2.Delete\n");
        printf("3.Display\n");
        printf("4.Quit\n");
        printf("buddy enter your choice : \n");
        scanf("%d", & choice);
        switch (choice)
        {
            case 1:
                printf("buddy type the element for insertion in queue : \n");
                scanf("%d", & items);
                Insert(items);
                break;
            case 2:
                Deletion();
                break;
            case 3:
                Display();
                break;
            case 4:
                break;
            default:
            printf("error 404 just kidding wrong choice select again\n");
        }
    }
    while (choice != 4);
    return 0;
}
```

Output:



```
1.Insert
2.Delete
3.Display
4.Quit
buddy enter your choice :
1
buddy type the element for insertion in queue :
23
1.Insert
2.Delete
3.Display
4.Quit
buddy enter your choice :
1
buddy type the element for insertion in queue :
46
1.Insert
2.Delete
3.Display
4.Quit
buddy enter your choice :
3
The elements in the Queue are:23 46
1.Insert
2.Delete
3.Display
4.Quit
```



```
buddy enter your choice :
3
The elements in the Queue are:23 46
1.Insert
2.Delete
3.Display
4.Quit
buddy enter your choice :
2
 The element deleted from queue is : 23
1.Insert
2.Delete
3.Display
4.Quit
buddy enter your choice :
5
error 404 just kidding wrong choice select again
1.Insert
2.Delete
3.Display
4.Quit
buddy enter your choice :
4


...Program finished with exit code 0
Press ENTER to exit console.
```