Question 1

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

#include <stdio.h>

# define max 6

**int** queue[max];  // array declaration

**int** front=-1;

**int** rear=-1;

1. // function to insert an element in a circular queue
2. **void** enqueue(**int** element)
3. {
4. **if**(front==-1 && rear==-1)   // condition to check queue is empty
5. {
6. front=0;
7. rear=0;
8. queue[rear]=element;
9. }
10. **else** **if**((rear+1)%max==front)  // condition to check queue is full
11. {
12. printf("Queue is overflow..");
13. }
14. **else**
15. {
16. rear=(rear+1)%max;       // rear is incremented
17. queue[rear]=element;     // assigning a value to the queue at the rear position.
18. }
19. }
21. // function to delete the element from the queue
22. **int** dequeue()
23. {
24. **if**((front==-1) && (rear==-1))  // condition to check queue is empty
25. {
26. printf("\nQueue is underflow..");
27. }
28. **else** **if**(front==rear)
29. {
30. printf("\nThe dequeued element is %d", queue[front]);
31. front=-1;
32. rear=-1;
33. }
34. **else**
35. {
36. printf("\nThe dequeued element is %d", queue[front]);
37. front=(front+1)%max;
38. }
39. }
40. // function to display the elements of a queue
41. **void** display()
42. {
43. **int** i=front;
44. **if**(front==-1 && rear==-1)
45. {
46. printf("\n Queue is empty..");
47. }
48. **else**
49. {
50. printf("\nElements in a Queue are :");
51. **while**(i<=rear)
52. {
53. printf("%d,", queue[i]);
54. i=(i+1)%max;
55. }
56. }
57. }
58. **int** main()
59. {
60. **int** choice=1,x;   // variables declaration
62. **while**(choice<4 && choice!=0)   // while loop
63. {
64. printf("\n Press 1: Insert an element");
65. printf("\nPress 2: Delete an element");
66. printf("\nPress 3: Display the element");
67. printf("\nEnter your choice");
68. scanf("%d", &choice);
70. **switch**(choice)
71. {
73. **case** 1:
75. printf("Enter the element which is to be inserted");
76. scanf("%d", &x);
77. enqueue(x);
78. **break**;
79. **case** 2:
80. dequeue();
81. **break**;
82. **case** 3:
83. display();

    }}

**return** 0;

}