#A Program To Implement Circular Queue Using Array.

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
int queue[SIZE];

int front = -1, rear = -1;

void enqueue(int element) {

   if ((rear + 1) % SIZE == front) {
      printf("\nQueue is full. Cannot insert %d.\n", element);
   } else {

      if (front == -1 && rear == -1) {
         front = rear = 0;
      } else {

         rear = (rear + 1) % SIZE;
      }
      queue[rear] = element;  // Insert the element
      printf("%d inserted into the queue.\n", element);
   }
}

int dequeue() {
   int dequeued;

   if (front == -1) {
      printf("\nQueue is empty. Cannot dequeue.\n");
      return -1;
```

```
    } else {
        dequeued = queue[front];

        if (front == rear) {
            front = rear = -1;
        } else {

            front = (front + 1) % SIZE;
        }
        printf("\nThe dequeued element is %d\n", dequeued);
        return dequeued;
    }
}

void display() {

    if (front == -1) {
        printf("\nQueue is empty.\n");
    } else {
        printf("\nElements in the queue are: ");
        int i = front;
        while (i != rear) {
            printf("%d ", queue[i]);
            i = (i + 1) % SIZE;
        }
        printf("%d\n", queue[rear]);  // Print the rear element
    }
}

int main() {
    int choice = 1, x;
```

```
    while (choice < 4 && choice != 0) {
        printf("\nPress 1: Insert an element");
        printf("\nPress 2: Delete an element");
        printf("\nPress 3: Display the elements");
        printf("\nEnter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter the element to be inserted: ");
                scanf("%d", &x);
                enqueue(x);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            default:
                printf("\nInvalid choice\n");
        }
    }
    return 0;
}
```