Rajalakshmi Engineering College

Name: Karan S
Email: 241801116@rajalakshmi.edu.in
Roll no: 241801116
Phone: 7339188168
Branch: REC
Department: I AI & DS FB
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_COD_Question 3

Attempt : 2
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Write a program to implement a queue using an array and pointers. The program should provide the following functionalities:

Insert an element into the queue.Delete an element from the queue.Display the elements in the queue.

The queue has a maximum capacity of 5 elements. If the queue is full and an insertion is attempted, a "Queue is full" message should be displayed. If the queue is empty and a deletion is attempted, a "Queue is empty" message should be displayed.

Input Format

Each line contains an integer representing the chosen option from 1 to 3.

Option 1: Insert an element into the queue followed by an integer representing the element to be inserted, separated by a space.

Option 2: Delete an element from the queue.

Option 3: Display the elements in the queue.

Output Format

For option 1 (insertion):-

1. The program outputs: "<data> is inserted in the queue." if the data is successfully inserted.
2. "Queue is full." if the queue is already full and cannot accept more elements.

For option 2 (deletion):-

1. The program outputs: "Deleted number is: <data>" if an element is successfully deleted and returns the value of the deleted element.
2. "Queue is empty." if the queue is empty no elements can be deleted.

For option 3 (display):-

1. The program outputs: "Elements in the queue are: <element1> <element2> ... <elementN>" where <element1>, <element2>, ..., <elementN> represent the elements present in the queue.
2. "Queue is empty." if the queue is empty no elements can be displayed.

For invalid options, the program outputs: "Invalid option."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 10

```
3
5
Output: 10 is inserted in the queue.
Elements in the queue are: 10
Invalid option.
```

Answer

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

#define max 5

int queue[max];
int front = -1, rear = -1;


int insertq(int *data) {
   if (rear == max - 1) {
      printf("Queue is full.\n");
      return 0;  // Queue is full
   }
   if (front == -1) {  // First element being inserted
      front = 0;
   }
   queue[++rear] = *data;
   printf("%d is inserted in the queue.\n", *data);
   return 1;
}

// Function to delete an element from the queue
int delq() {
   if (front == -1 || front > rear) {  // Queue is empty
      return -1;
   }
   int deleted = queue[front++];  // Remove the front item
   if (front > rear) {
      front = rear = -1;  // Reset queue if empty
   }
   printf("Deleted number is: %d\n", deleted);  // Print the deleted number
   return deleted;
}
```

```
// Function to display the elements of the queue
void display() {
    if (front == -1 || front > rear) {  // Queue is empty
        printf("Queue is empty.\n");
    } else {
        printf("Elements in the queue are:");
        for (int i = front; i <= rear; i++) {
            printf(" %d", queue[i]);
        }
        printf("\n");
    }
}

int main()
{
    int data, reply, option;
    while (1)
    {
        if (scanf("%d", &option) != 1)
            break;
        switch (option)
        {
            case 1:
                if (scanf("%d", &data) != 1)
                    break;
                reply = insertq(&data);
                if (reply == 0)
                    printf("Queue is full.\n");
                else
                    printf("%d is inserted in the queue.\n", data);
                break;
            case 2:
                delq();  //    Called without arguments
                break;
            case 3:
                display();
                break;
            default:
                printf("Invalid option.\n");
                break;
        }
    }
```

```
    return 0;
}
```

Status : Wrong ***Marks : 0/10***