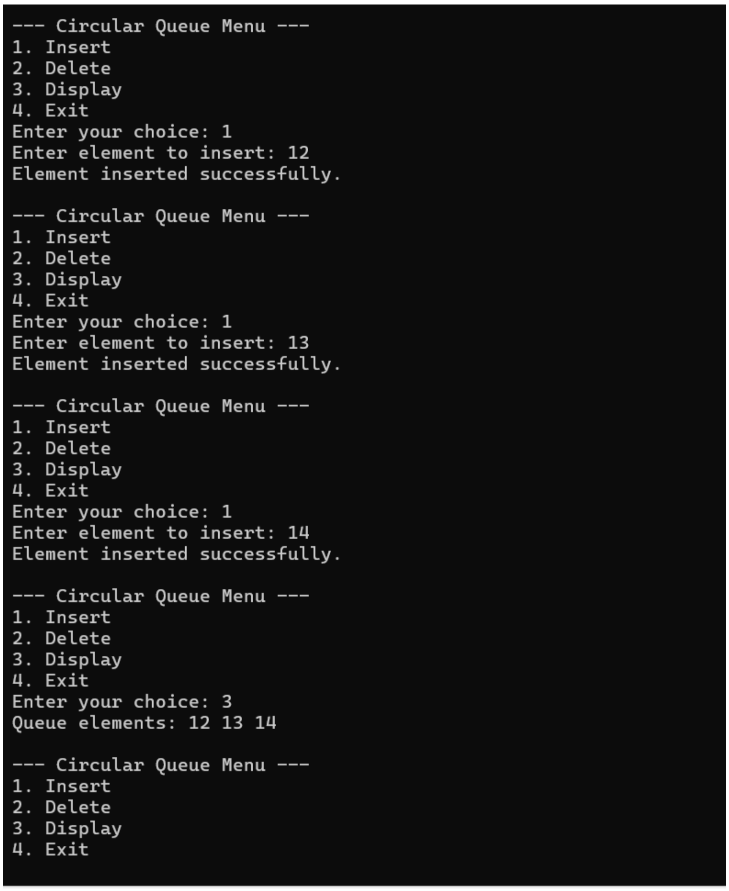
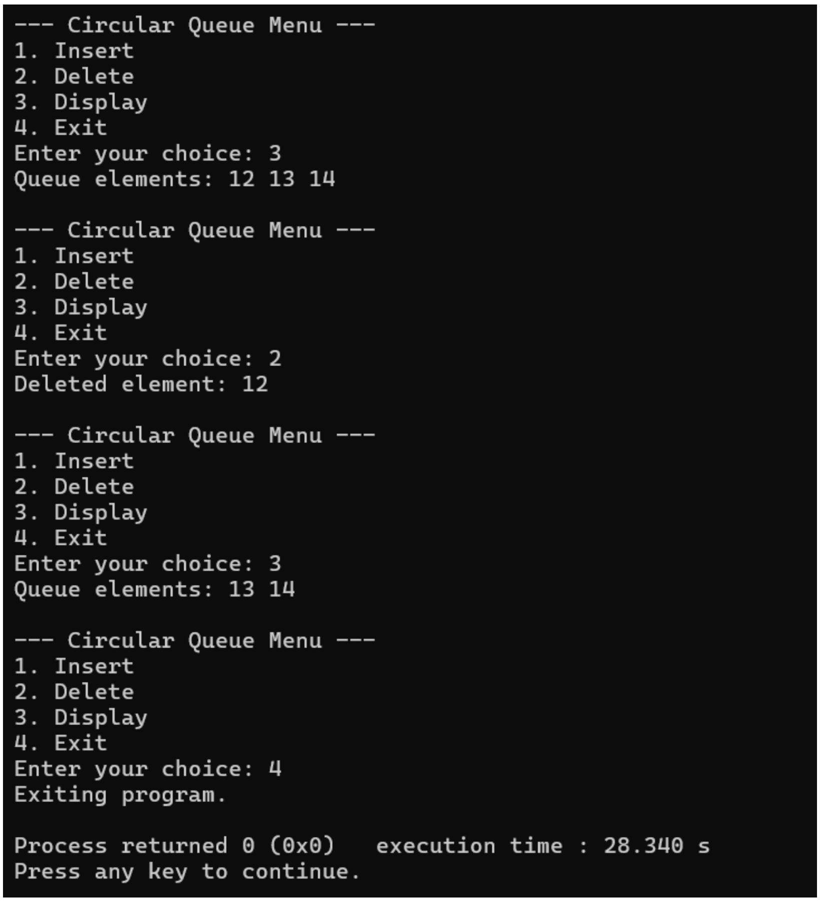
Lab program 3b:-

)WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display The program should print appropriate messages for queue empty and queue overflow conditions

```
#include <stdio.h>


#define MAX 5   // Size of circular queue


int cq[MAX];

int front = -1, rear = -1;


/* Insert operation */

void insert() {

    int item;

    if ((rear + 1) % MAX == front) {

        printf("Queue Overflow! Cannot insert element.\n");

        return;

    }


    printf("Enter element to insert: ");

    scanf("%d", &item);


    if (front == -1) {   // First element

        front = 0;

        rear = 0;

    } else {

        rear = (rear + 1) % MAX;

    }


    cq[rear] = item;
    printf("Element inserted successfully.\n");
}


/* Delete operation */
void delete() {
    if (front == -1) {
        printf("Queue Empty! Cannot delete element.\n");
        return;
    }

    printf("Deleted element: %d\n", cq[front]);

    if (front == rear) {   // Only one element
        front = -1;
        rear = -1;
    } else {
        front = (front + 1) % MAX;
    }
}


/* Display operation */
void display() {
    int i;

    if (front == -1) {
        printf("Queue Empty! Nothing to display.\n");
        return;
    }
    printf("Queue elements: ");
    i = front;
    while (1) {
        printf("%d ", cq[i]);
        if (i == rear)
            break;
        i = (i + 1) % MAX;
    }
    printf("\n");
}

/* Main function */
int main() {
    int choice;

    do {
        printf("\n--- Circular Queue Menu ---\n");
        printf("1. Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                insert();
                break;
            case 2:
                delete();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                break;
            default:
                printf("Invalid choice! Try again.\n");
        }
    } while (choice != 4);

    return 0;
}
```



```
--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 12
Element inserted successfully.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 13
Element inserted successfully.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter element to insert: 14
Element inserted successfully.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 12 13 14

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
```



```
--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 12 13 14

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted element: 12

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 13 14

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting program.

Process returned 0 (0x0)   execution time : 28.340 s
Press any key to continue.
```