1). WAP to simulate the working of a queue of integers using an array. Provide the following operations

a) Insert

b) Delete

c) Display

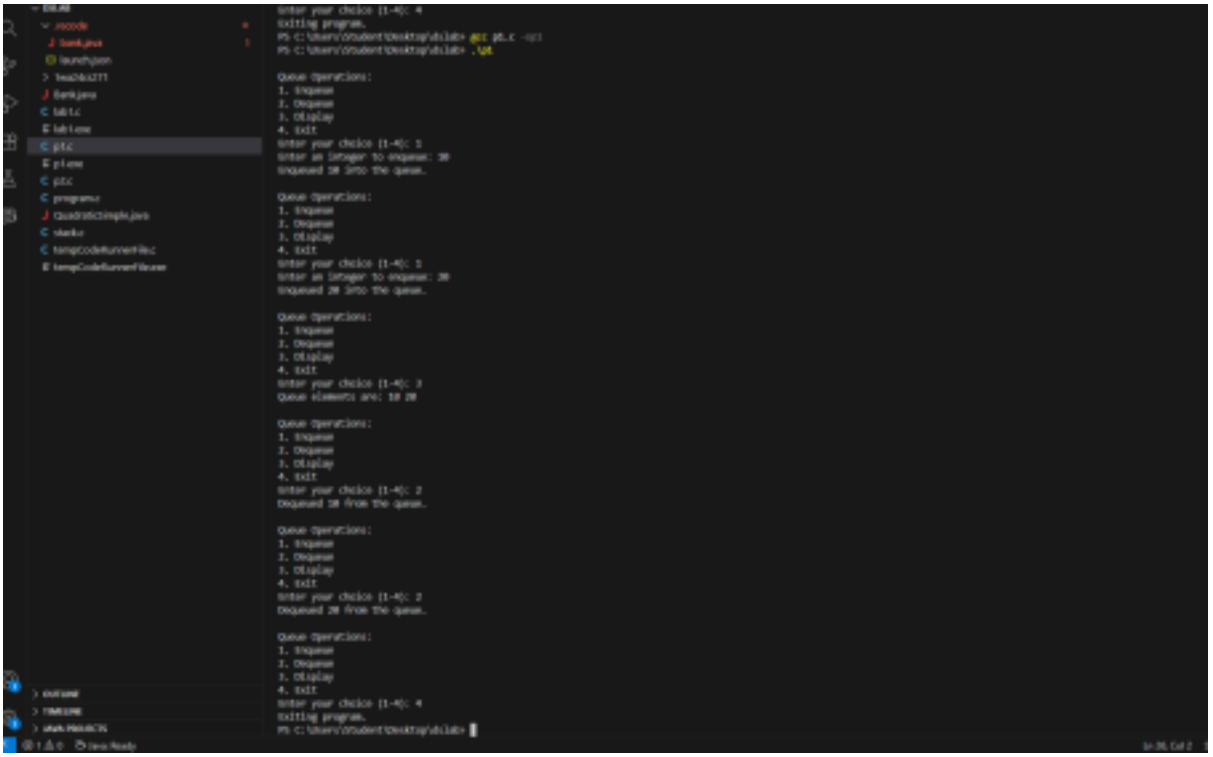
The program should print appropriate messages for queue empty and queue overflow conditions

```c
#include <stdio.h>
#define MAX 5

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

int isFull() {
    return rear == MAX - 1;
}

int isEmpty() {
    return front == -1 || front > rear;
}

void enqueue(int value) {
    if (isFull()) {
        printf("Queue Overflow! Cannot enqueue %d\n",
        value); return;
    }
    if (front == -1) {
        front = 0;
    }
    rear++;
    queue[rear] = value;
    printf("Enqueued %d into the queue.\n", value);
}

void dequeue() {
    if (isEmpty()) {
        printf("Queue Underflow! Queue is empty.\n");
        return;
    }
    int dequeuedValue = queue[front];
    front++;
    if (front > rear) {
```

```
        front = -1;
        rear = -1;
    }
    printf("Dequeued %d from the queue.\n",
dequeuedValue); }

void display() {
    if (isEmpty()) {
        printf("Queue is empty.\n");
        return;
    }
    printf("Queue elements are: ");
    for (int i = front; i <= rear; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");
}

int main() {
    int choice, value;

    while (1) {
        printf("\nQueue Operations:\n");
        printf("1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Display\n");
        printf("4. Exit\n");

        printf("Enter your choice (1-4): ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter an integer to enqueue: ");
                scanf("%d", &value);
                enqueue(value);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
```

```
            return 0;
        default:
            printf("Invalid choice. Please enter 1-4.\n");
        }
    }
    return 0;
}
```

o/p:-



2).WAP to simulate the working of a circular queue of integers using an array. Provide the following operations.

a) Insert

b) Delete

c) Display

The program should print appropriate messages for queue empty and queue overflow

Conditions

```c
#include <stdio.h>
#define MAX 5

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

int isFull() {
    return (front == (rear + 1) % MAX);
}

int isEmpty() {
    return (front == -1);
}

void enqueue(int value) {
    if (isFull()) {
        printf("Queue Overflow! Cannot enqueue %d\n",
        value); return;
    }
    if (isEmpty()) {
        front = 0;
    }
    rear = (rear + 1) % MAX;
    queue[rear] = value;
    printf("Enqueued %d into the queue.\n",
value); }

void dequeue() {
    if (isEmpty()) {
        printf("Queue Underflow! Queue is empty.\n");
        return;
    }
    int dequeuedValue = queue[front];
    if (front == rear) {

        front = -1;
        rear = -1;
    } else {
        front = (front + 1) % MAX;
    }
    printf("Dequeued %d from the queue.\n",
dequeuedValue); }

void display() {
    if (isEmpty()) {
```

```
        printf("Queue is empty.\n");
        return;
    }
    printf("Queue elements are: ");
    int i = front;
    while (1) {
        printf("%d ", queue[i]);
        if (i == rear)
            break;
        i = (i + 1) % MAX;
    }
    printf("\n");
}

int main() {
    int choice, value;

    while (1) {
        printf("\nQueue Operations:\n");
        printf("1. Enqueue\n2. Dequeue\n3. Display\n4.
        Exit\n"); printf("Enter your choice (1-4): ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter an integer to enqueue: ");
                scanf("%d", &value);
                enqueue(value);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice. Please enter 1-4.\n");
        }
    }
    return 0;
}
```

o/p:-