**Name-Jayanth MM**

**Section-I**

**DSA: Lab Program-3**

**b) WAP to simulate the working of Circular Queue using an array with the following operations: Insert, Delete and Display, also should print appropriate message for queue empty and overflow conditions.**

#include <stdio.h>

#define SIZE 3

int queue[SIZE];

int front = -1, rear = -1;

void insert() {

int x;

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

printf("Queue Overflow!\n");

return;

}

printf("Enter value: ");

scanf("%d", &x);

if (front == -1) front = rear = 0;

else rear = (rear + 1) % SIZE;

queue[rear] = x;

printf("%d inserted.\n", x);

}

void delete() {

if (front == -1) {

printf("Queue Underflow!\n");

return;

}

printf("%d deleted.\n", queue[front]);

if (front == rear) front = rear = -1;

else front = (front + 1) % SIZE;

}

void display() {

if (front == -1) {

printf("Queue is empty.\n");

return;

}

printf("Queue: ");

int i = front;

while (1) {

printf("%d ", queue[i]);

if (i == rear) break;

i = (i + 1) % SIZE;

}

printf("\n");

}

int main() {

int choice;

printf("---Circular Queue---");

while (1) {

printf("\n1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");

scanf("%d", &choice);

switch(choice) {

case 1: insert(); break;

case 2: delete(); break;

case 3: display(); break;

case 4: return 0;

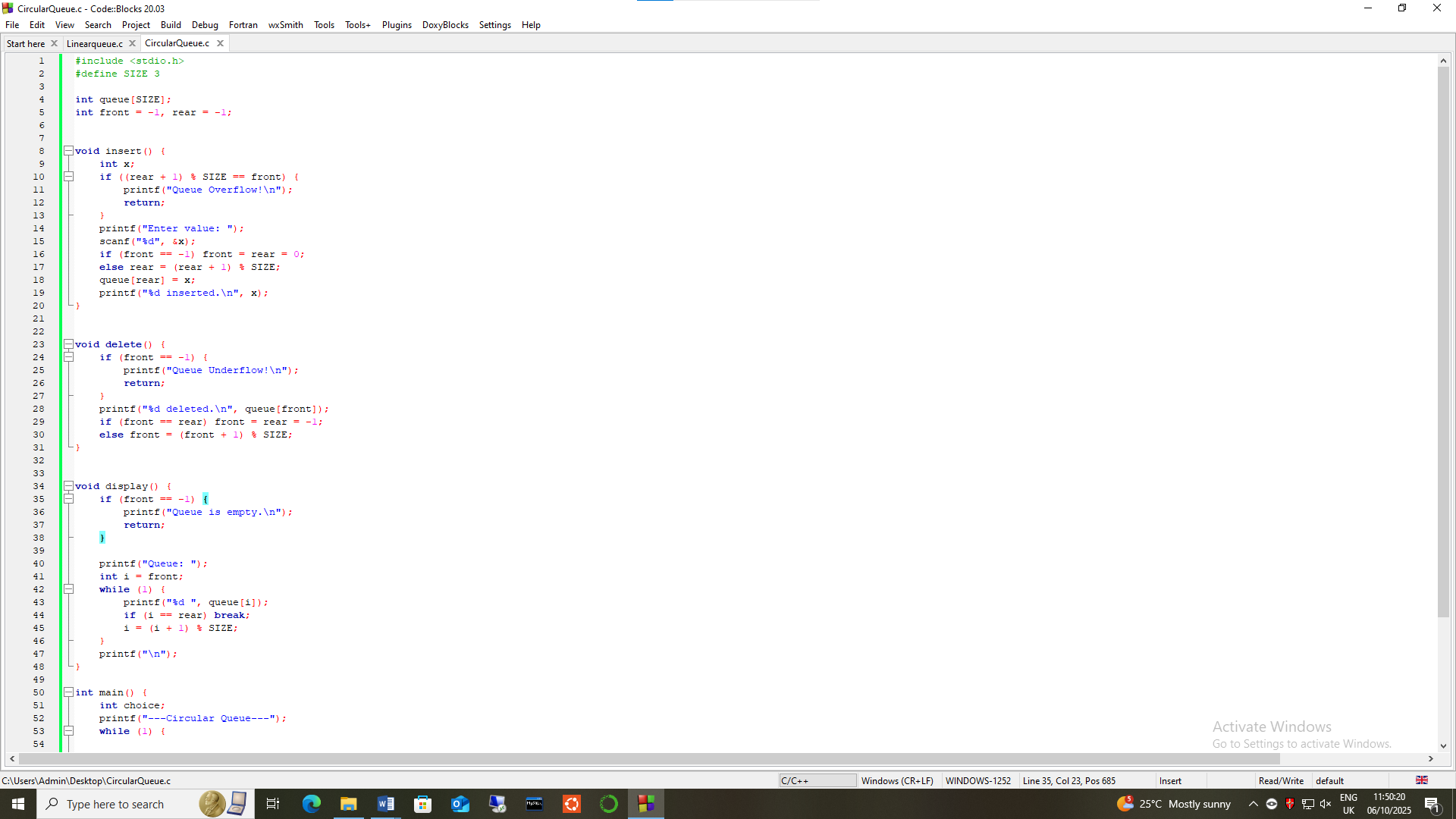
default: printf("Invalid choice!\n");

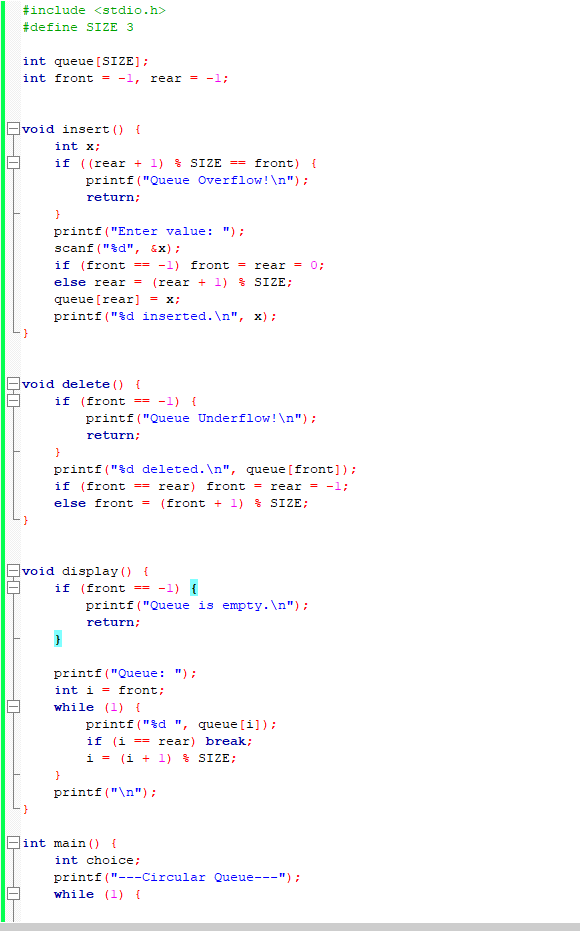
}

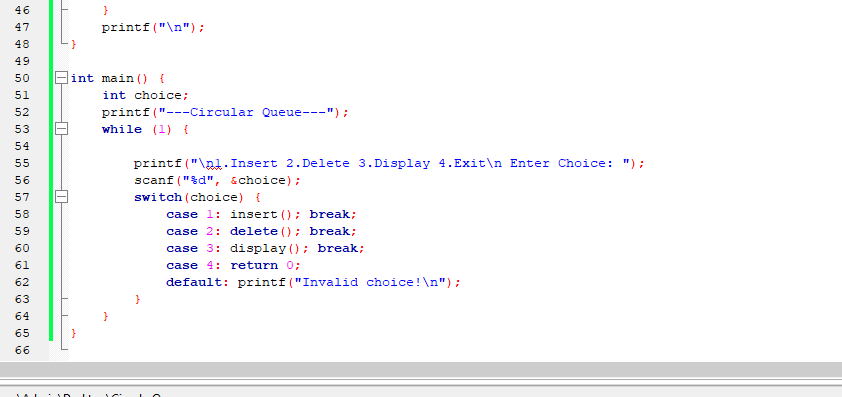
}

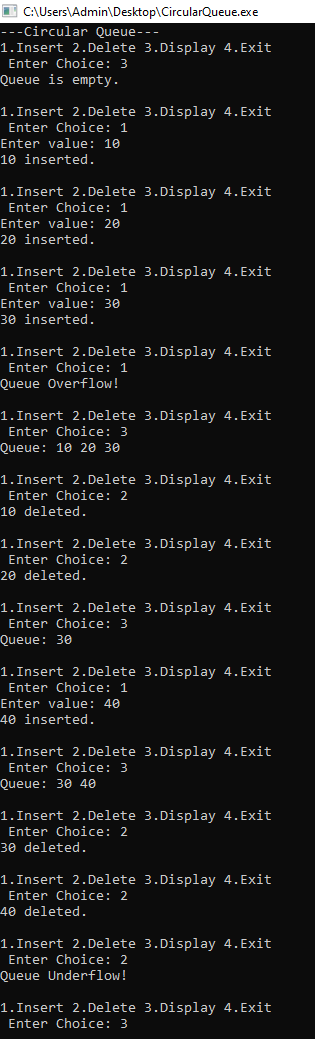
}

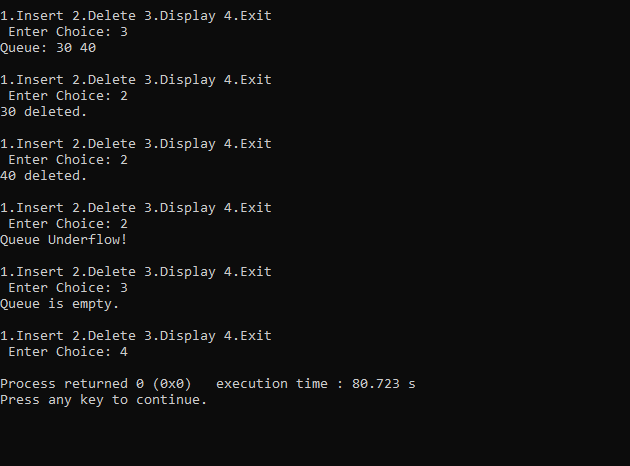
**Code and Expected Output:**

****

****

****

****

****