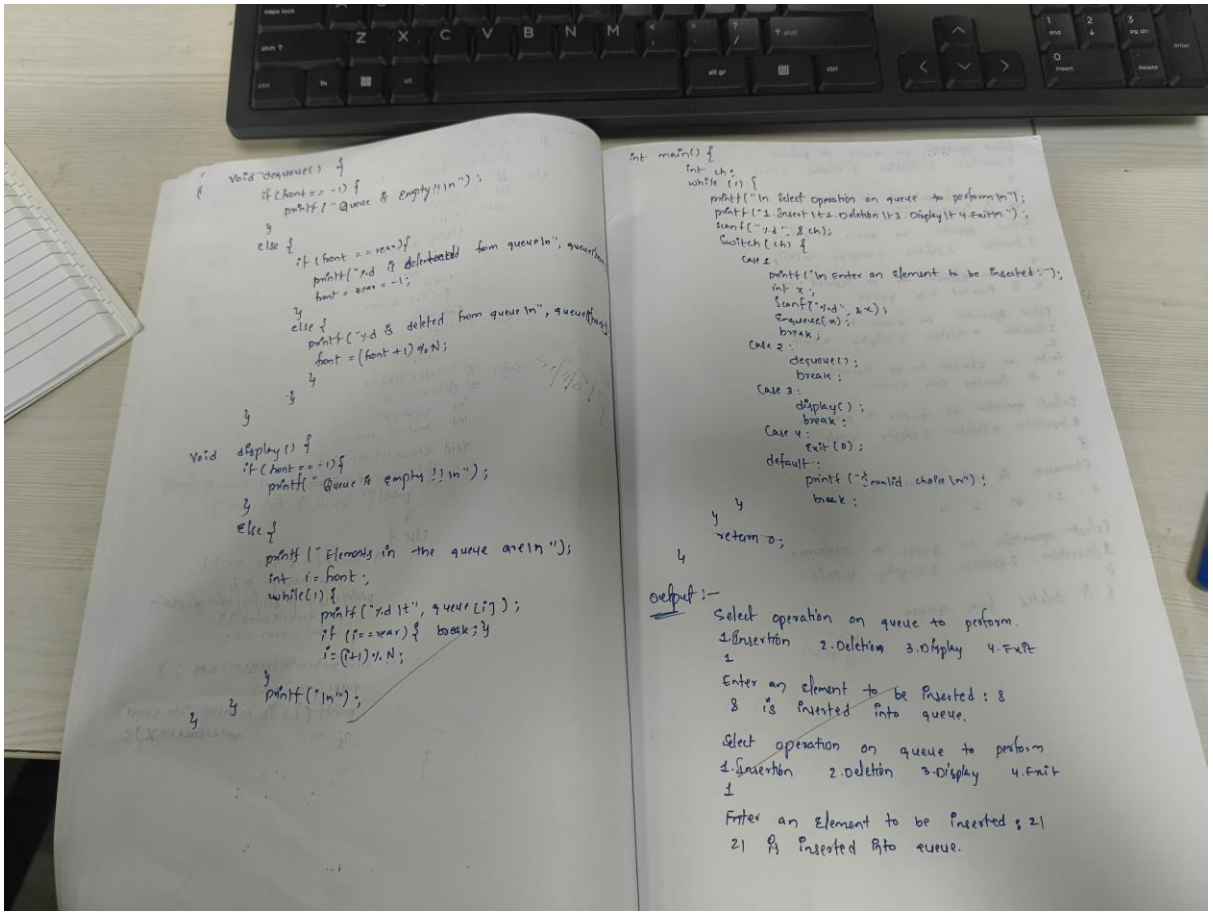
```c
#include<stdio.h>
#include<stdlib.h>
#define N 100
int stack[N];
int top=-1;
void display();
void push(){
    int l;
    printf("enter element to be inserted\n");
    scanf("%d",&l);
    if(top==N-1){printf("Stack overflow");}
    else{
        top++;
        stack[top]=l;
        printf("%d is pushed into stack\n",stack[top]);
        display();
    }
}
void pop(){
    if(top==-1){printf("Stack underflow");}
    else{
       int item =stack[top];
       top--;
       printf("%d is popped out of stack\n",item);
       display();
    }
}
void peek(){
    if(top==-1){printf("Stack underflow");}
    else{
        printf("top element in stack is:%d",stack[top]);
    }
}
void display(){
    if(top==-1){printf("empty stack");}
    else{
        printf("\nElements in the stack are:\t");
        for(int i=top;i>=0;i--){
            printf("%d\t",stack[i]);
        }
    }
}
```

```c
int main(){
    int ch;
    while(1){
        printf("\n1.Push\t2.Pop\t3.Peek\t4.Dislpay\t5.Exit\n");
        printf("Enter stack opertaion:\n");
        scanf("%d",&ch);
        switch(ch){
            case 1:
                push();
                break;
            case 2:
                pop();
                break;
            case 3:
                peek();
                break;
            case 4:
                display();
                break;
            case 5:
                exit(0);
            default:
                printf("Invalid choice");
                break;
            }
    }
    return 0;
}
```

```
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
1
enter element to be inserted
12
12 is pushed into stack

Elements in the stack are:      12
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
1
enter element to be inserted
2
2 is pushed into stack

Elements in the stack are:      2       12
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
2
2 is popped out of stack

Elements in the stack are:      12
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
3
top element in stack is:12
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
2
12 is popped out of stack
empty stack
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
2
Stack underflow
1.Push  2.Pop   3.Peek  4.Dislpay       5.Exit
Enter stack opertaion:
5

Process returned 0 (0x0)   execution time : 367.217 s
Press any key to continue.
```



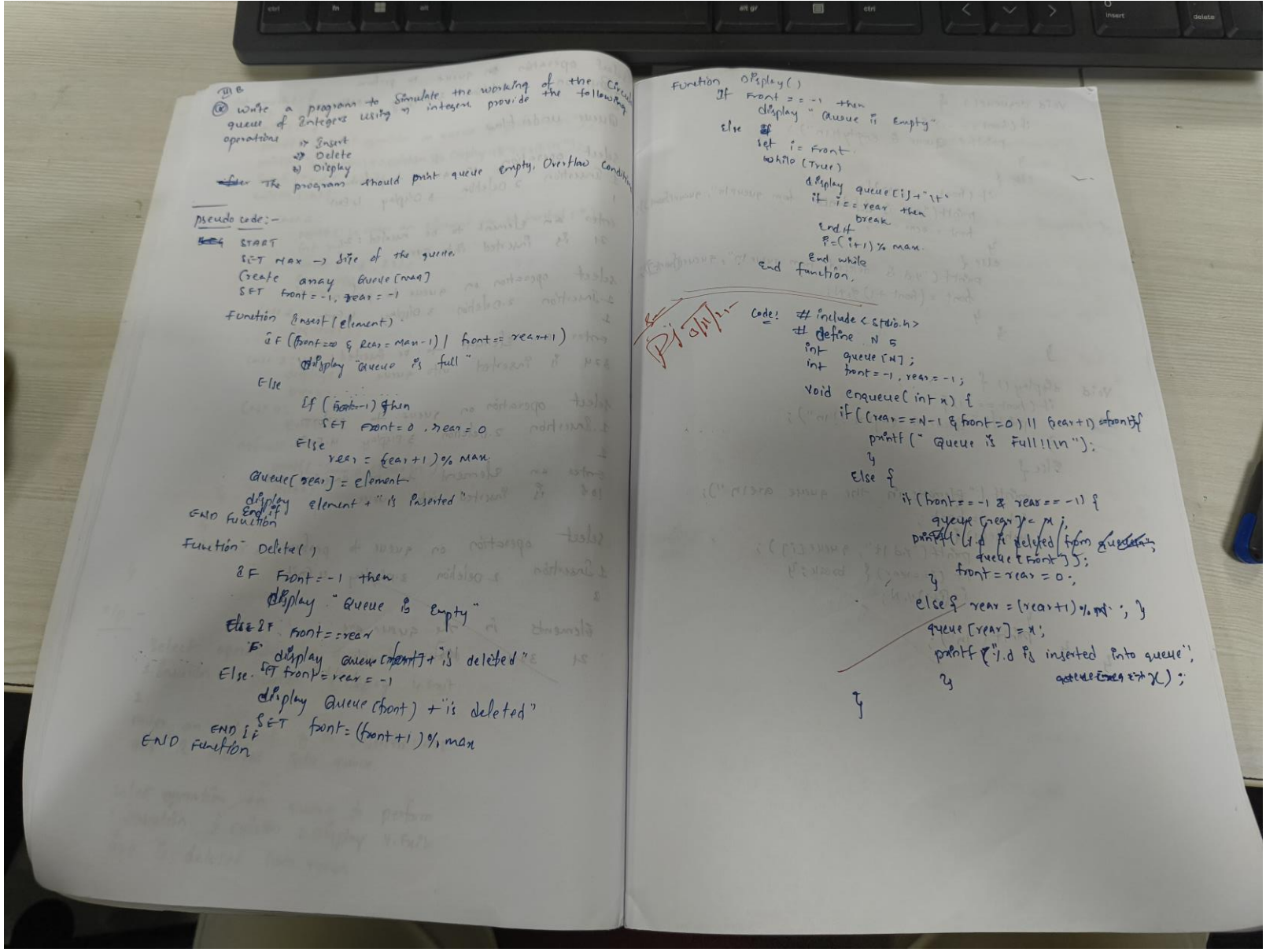
```
void dequeue() {
    if (front == -1) {
        printf("Queue is empty!!\n");
    }
    else {
        if (front == rear) {
            printf("%d is deleted from queue\n", queue[front]);
            front = rear = -1;
        }
        else {
            printf("%d is deleted from queue\n", queue[front]);
            front = (front + 1) % N;
        }
    }
}

void display() {
    if (front == -1) {
        printf("Queue is empty!!\n");
    }
    else {
        printf("Elements in the queue are\n");
        int i = front;
        while(1) {
            printf("%d\t", queue[i]);
            if (i == rear) { break; }
            i = (i+1) % N;
        }
        printf("\n\n");
    }
}
```

```
int main() {
    int ch;
    while (1) {
        printf("\n Select operation on queue to perform\n");
        printf("1.Insert\t2.Deletion\t3.Display\t4.Exit\n");
        scanf("%d", &ch);
        switch (ch) {
            case 1:
                printf("\n Enter an element to be Inserted:");
                int x;
                scanf("%d", &x);
                enqueue(x);
                break;
            case 2:
                dequeue();
                break;
            case 3:
                display();
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice\n");
                break;
        }
    }
    return 0;
}
```

output :-

Select operation on queue to perform.
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be Inserted : 8
8 is inserted into queue.

Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be inserted : 21
21 is inserted into queue.



Q6 (a) Write a program to simulate the working of the circular queue of integers using an integer provide the following operations
i) Insert
ii) Delete
iii) Display
The program should print queue empty, Overflow condition

Pseudo code:-

START
SET MAX → size of the queue
Create array Queue[max]
SET front = -1, rear = -1

Function Insert(element)
IF (front==0 & Rear = max-1) || front == rear+1)
display "Queue is full"
Else
If (front==-1) then
SET Front=0, rear=0
Else
rear = (rear+1)% MAX
Queue[rear] = element
display element + "is Inserted"
End if
END Function

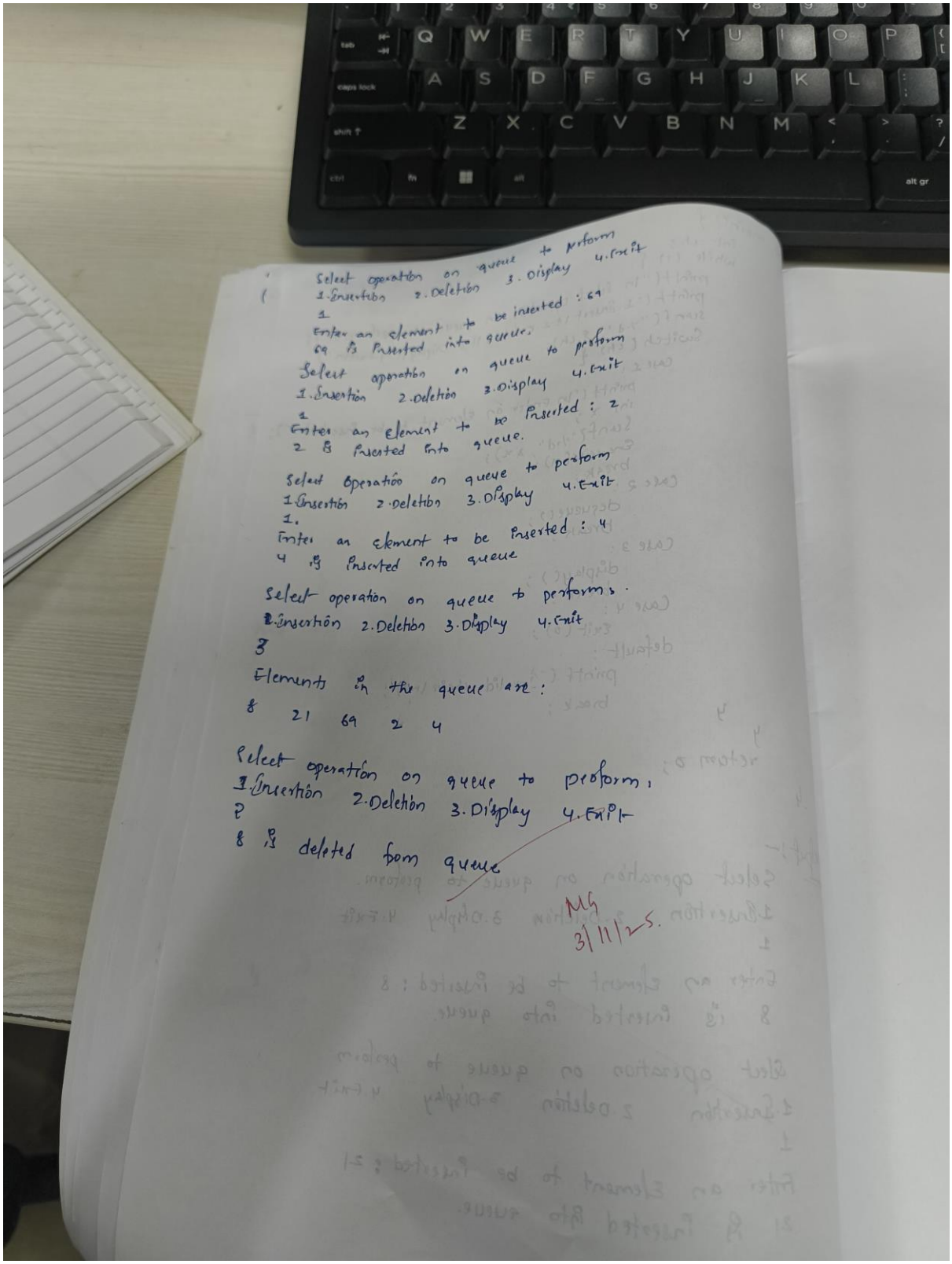
Function Delete()
IF Front==-1 then
display "Queue is Empty"
Else IF front==rear
display Queue[front] + "is deleted"
Else SET front=rear=-1
display Queue(front) + "is deleted"
SET front=(front+1)%max
END IF
END Function

Function Display()
If Front == -1 then
display "Queue is Empty"
else
let i = Front
while (True)
display queue[i] + "it"
if i == rear then
break
end if
i=(i+1)% max
end while
end function

Pgm:-

Code:-

```
#include <stdio.h>
#define N 5
int queue[N];
int front=-1, rear=-1;
void enqueue(int x) {
if((rear==N-1 && front=0) || (rear+1)==front){
printf("Queue is Full!!!\n");
}
else {
if(front==-1 && rear==-1) {
queue[rear]=x;
printf("%d is deleted from queue",
queue[front]);
front=rear=0;
}
else rear=(rear+1)%N; }
queue[rear]=x;
printf("%d is inserted into queue",
}
```


Select operation on queue to perform
1.Insertion 2.Deletion 3. Display 4.Exit
1
Enter an element to be inserted : 69
69 is Inserted into queue.
Select operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1
Enter an element to be Inserted : 2
2 is Inserted into queue.
Select Operation on queue to perform
1.Insertion 2.Deletion 3.Display 4.Exit
1.
Enter an element to be Inserted : 4
4 is Inserted into queue
Select operation on queue to perform:
1.Insertion 2.Deletion 3.Display 4.Exit
3
Elements in the queue are :
8 21 69 2 4
Select operation on queue to proform:
1.Insertion 2.Deletion 3.Display 4.Exit
2
8 is deleted from queue
MG
3/11/25.

C