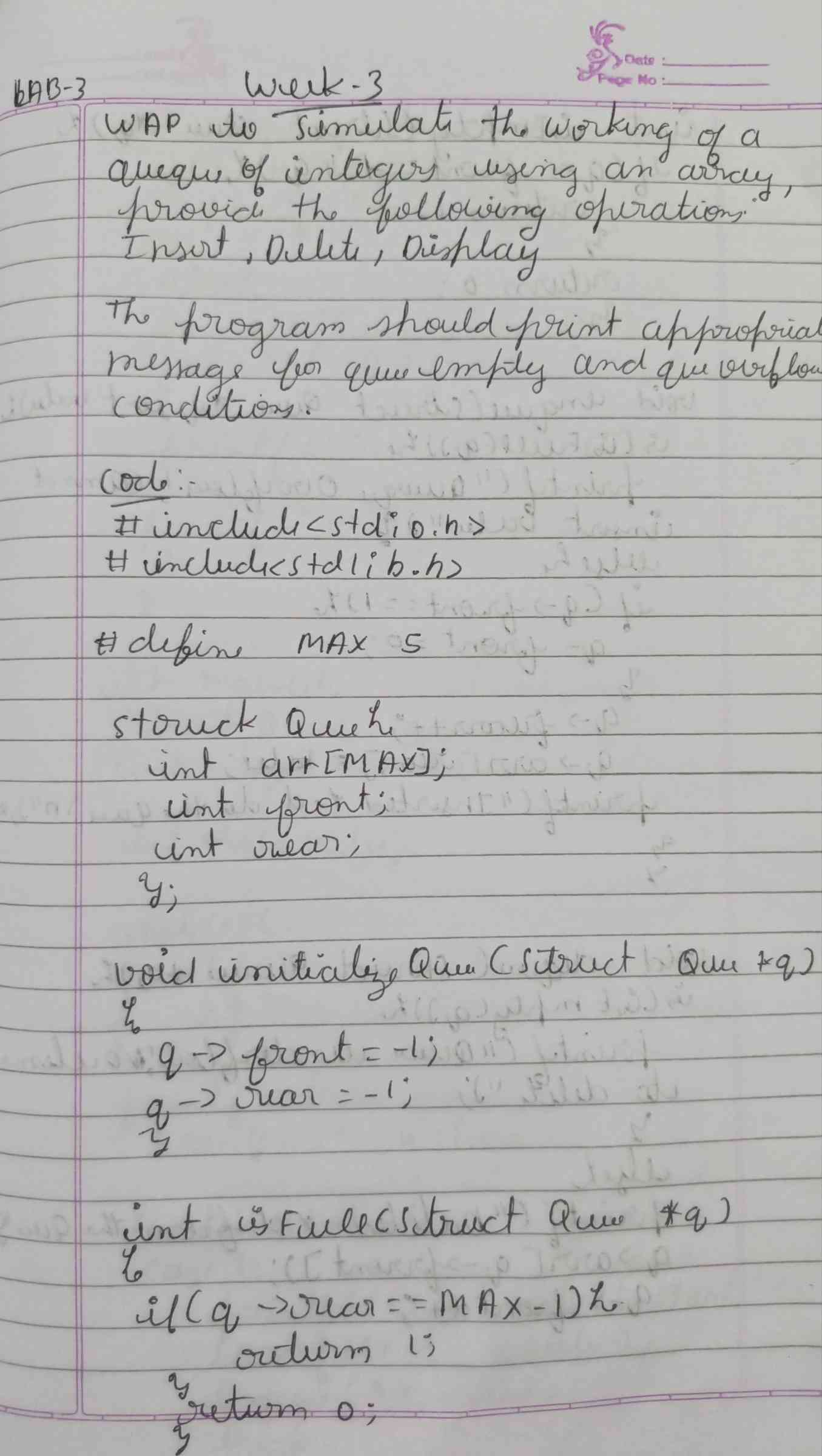
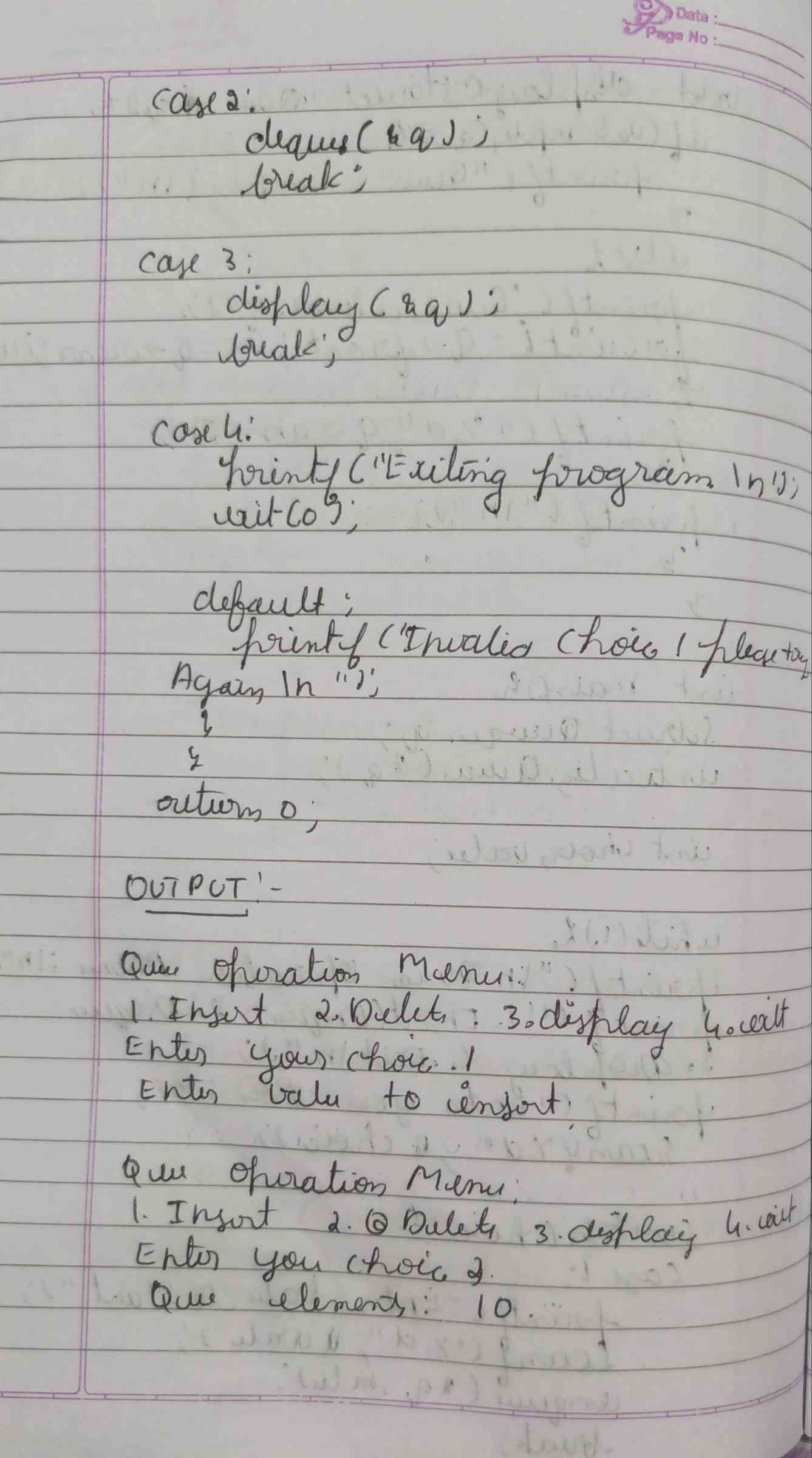
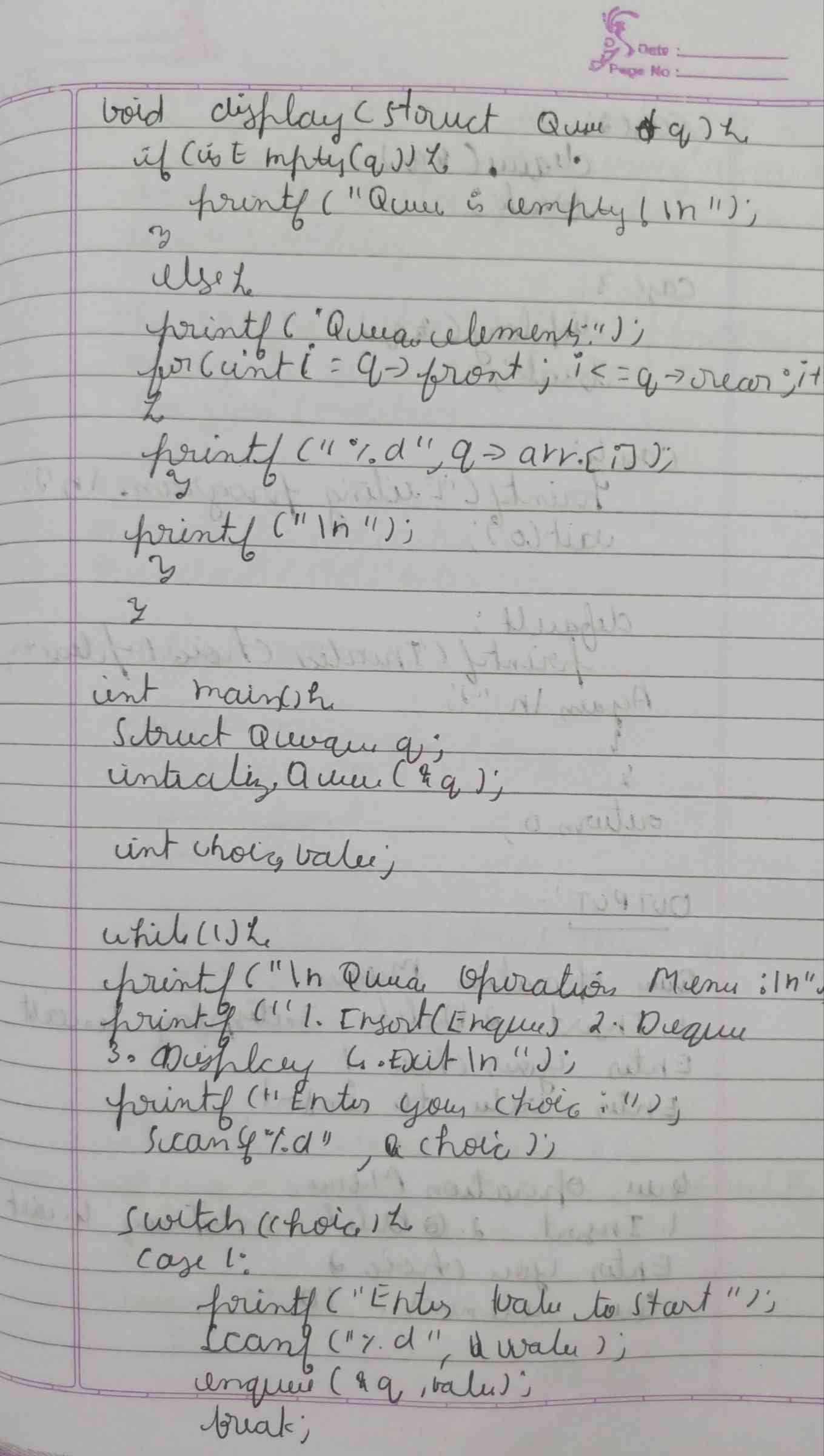
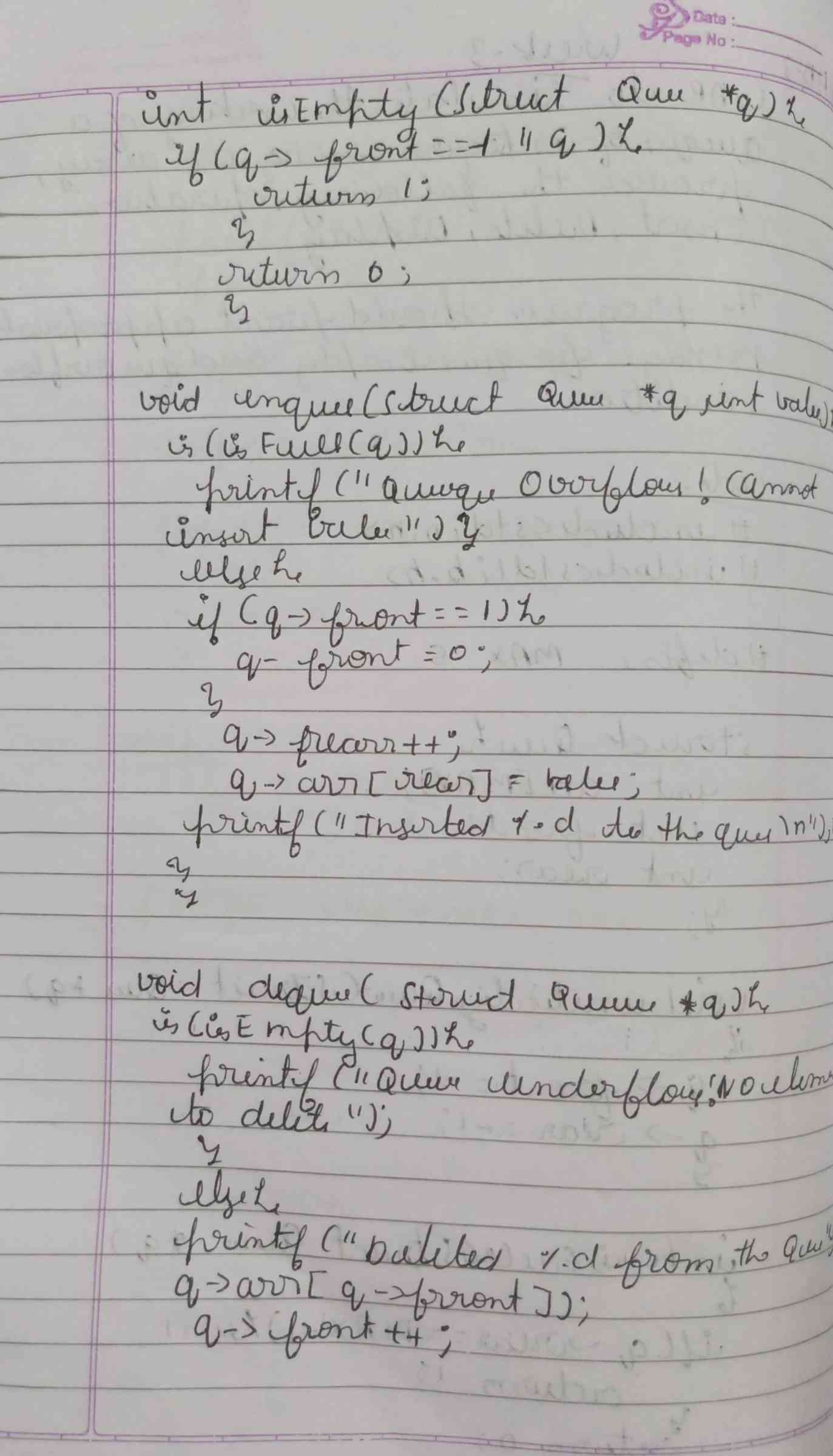
**LABORATORY PROGRAM – 3**

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

**OBSERVATION :**

****

****

**CODE :**

**#include <stdio.h>**

**#include <stdlib.h>**

**#define MAX 5**

**int isFull(int rear) {**

**if (rear == MAX - 1) {**

**return 1;**

**}**

**return 0; }**

**int isEmpty(int front, int rear) {**

**if (front == -1 || front > rear) {**

**return 1;**

**}**

**return 0; }**

**void insert(int queue[], int \*front, int \*rear, int value) {**

**if (isFull(\*rear)) {**

**printf("Queue Overflow! Cannot insert %d\n", value);**

**return;**

**}**

**if (\*front == -1) {**

**\*front = 0;**

**}**

**(\*rear)++;**

**queue[\*rear] = value;**

**printf("%d inserted into the queue\n", value);**

**}**

**void delete(int queue[], int \*front, int \*rear) {**

**if (isEmpty(\*front, \*rear)) {**

**printf("Queue Underflow! No element to delete\n");**

**return;**

**}**

**int deletedValue = queue[\*front];**

**printf("%d deleted from the queue\n", deletedValue);**

**(\*front)++;**

**if (\*front > \*rear) {**

**\*front = \*rear = -1;**

**}**

**}**

**void display(int queue[], int front, int rear) {**

**if (isEmpty(front, rear)) {**

**printf("Queue is empty!\n");**

**return;    }**

**printf("Queue elements: ");**

**for (int i = front; i <= rear; i++) {**

**printf("%d ", queue[i]);  }**

**printf("\n");**

**}**

**int main() {**

**int queue[MAX];**

**int front = -1, rear = -1;**

**int choice, value;**

**while (1) {**

**printf("\nQueue Operations:\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:**

**printf("Enter the value to insert: ");**

**scanf("%d", &value);**

**insert(queue, &front, &rear, value);**

**break;**

**case 2:**

**delete(queue, &front, &rear);**

**break;**

**case 3:**

**display(queue, front, rear);**

**break;**

**case 4:**

**exit(0);**

**default:**

**printf("Invalid choice! Please try again.\n");**

**}    }**

**return 0;**

**}**

**OUTPUT :**

