

**William Carey**

**C16315253**

---

**Algorithms and  
problem solving  
Project Report**

**Course Code**

**DT-228**

**Year 1**

**Computer Science**

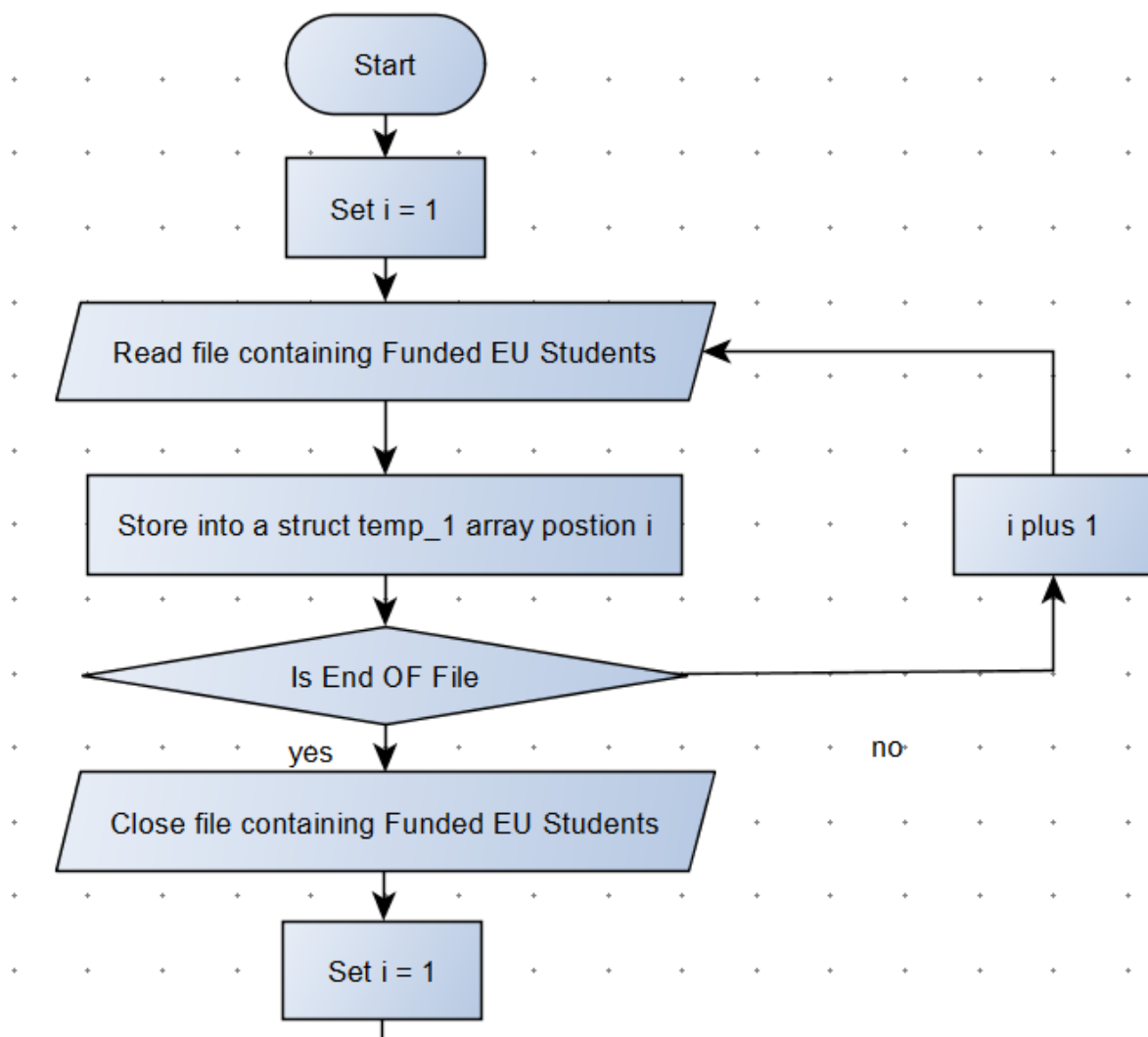
---

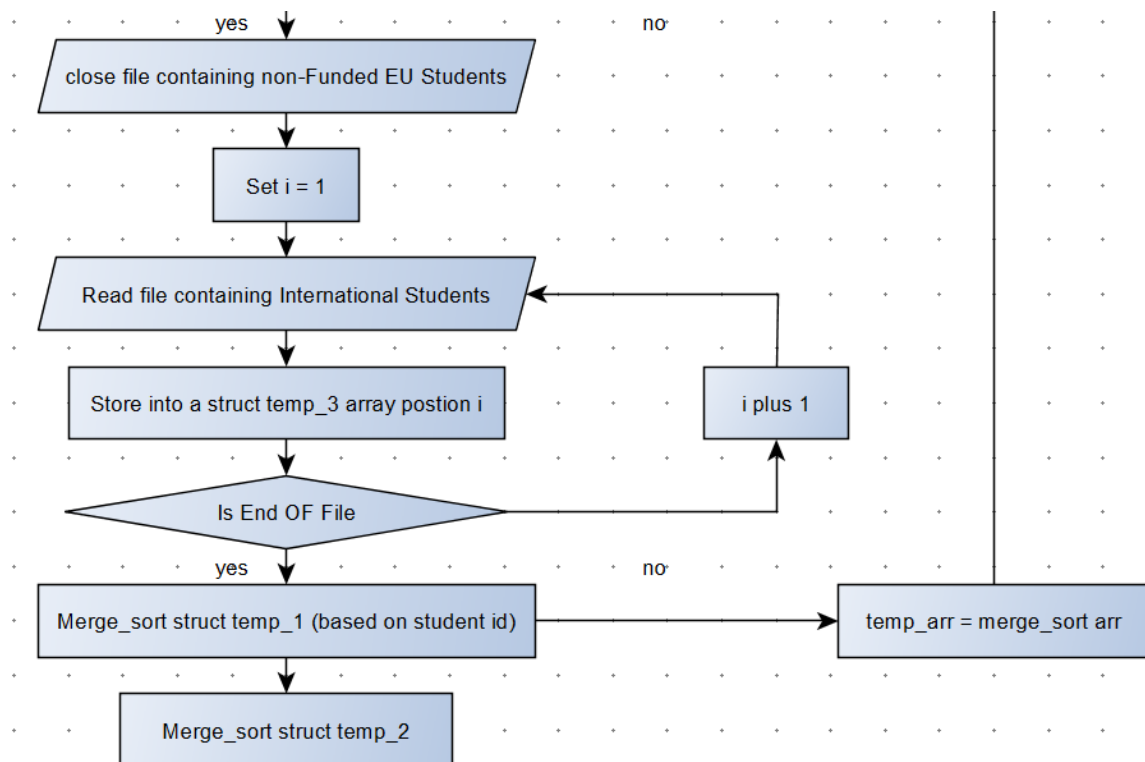
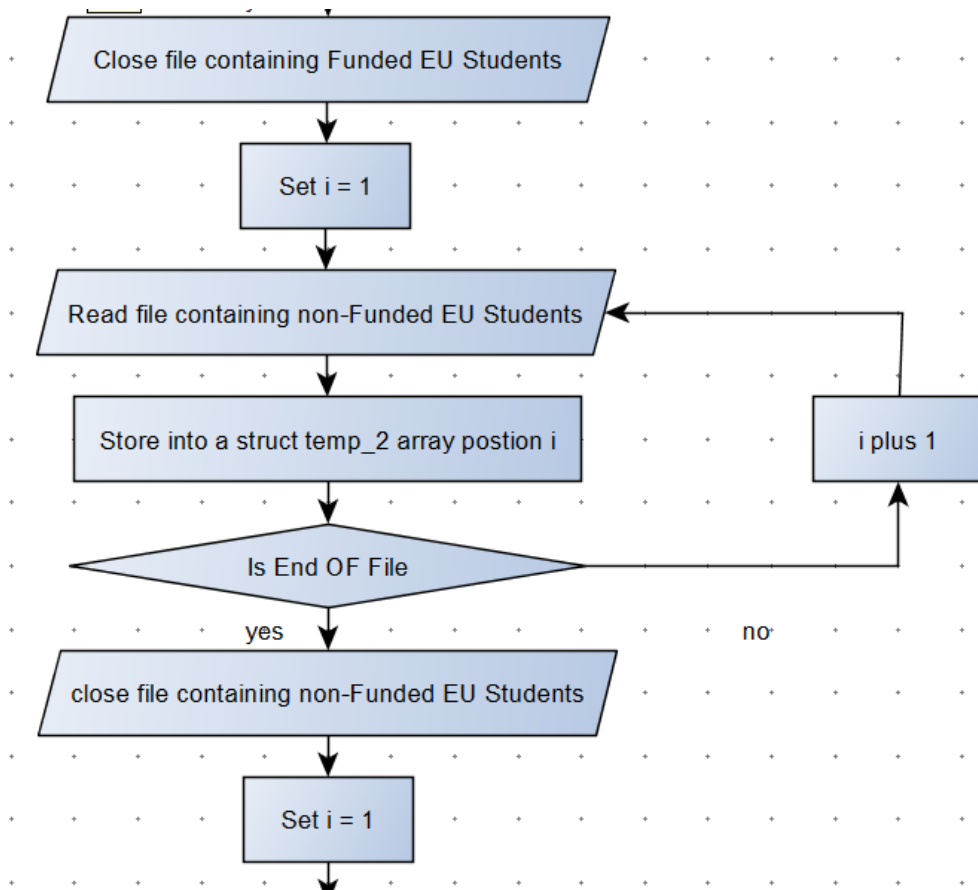
## Contents

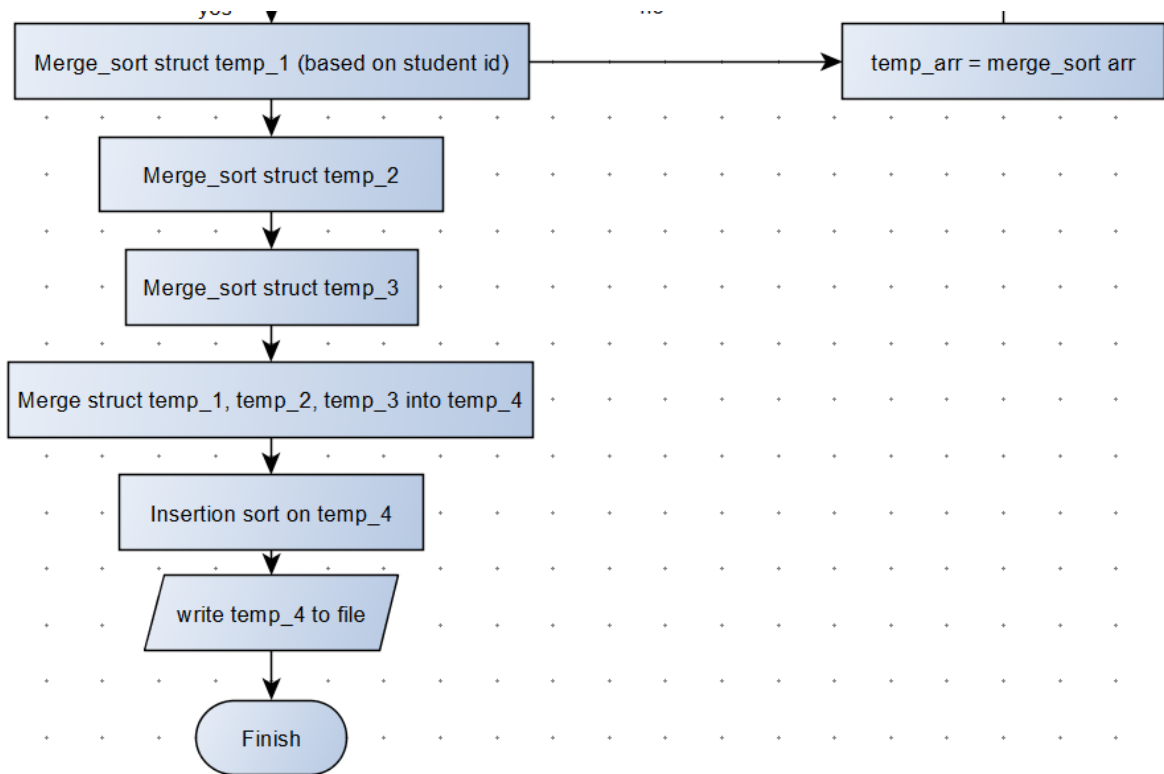
Question 1 – Flowchart that sorts out three list into one .....	3
Question 2 pseudo-code on searching for international students .....	8
Question 3 pseudo-code on searching for a particular student by surname .....	9

## Question 1 – Flowchart that sorts out three list into one

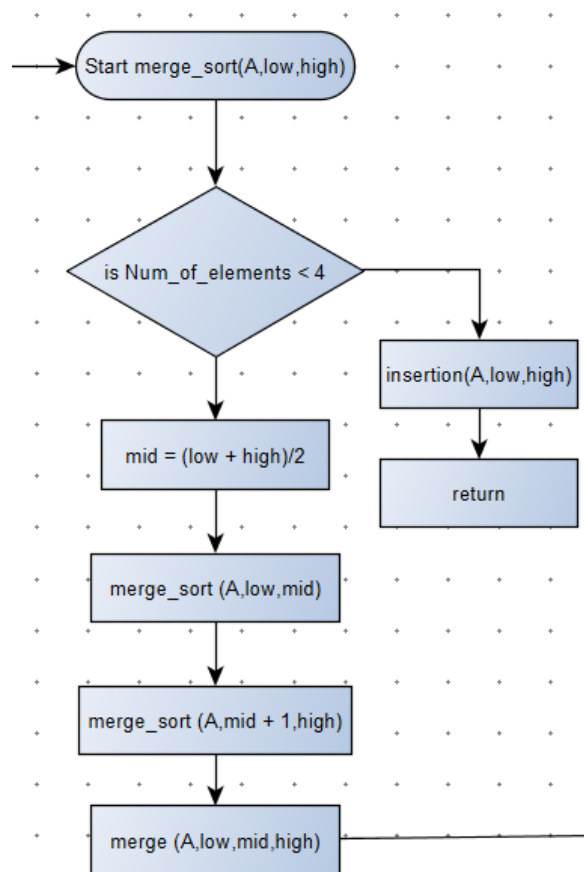
Within the merge sort, there is a insertion sort. The base case was changed to “if  $N < 8$ ” to allow this change. Because of this, the big O of this algorithm is  $O((N^3)(\log(N)))$  as the insertion sort has the big O of  $O(N^2)$  whereas the merge sort has a worst case of  $O(N(\log(N)))$ .  $(N \times N^2) = (N^3)$ . This means within the algorithm there is a loop within a loop within a loop. All the students are sorted by student ID as that between all the student is the only thing that is unique between each student.

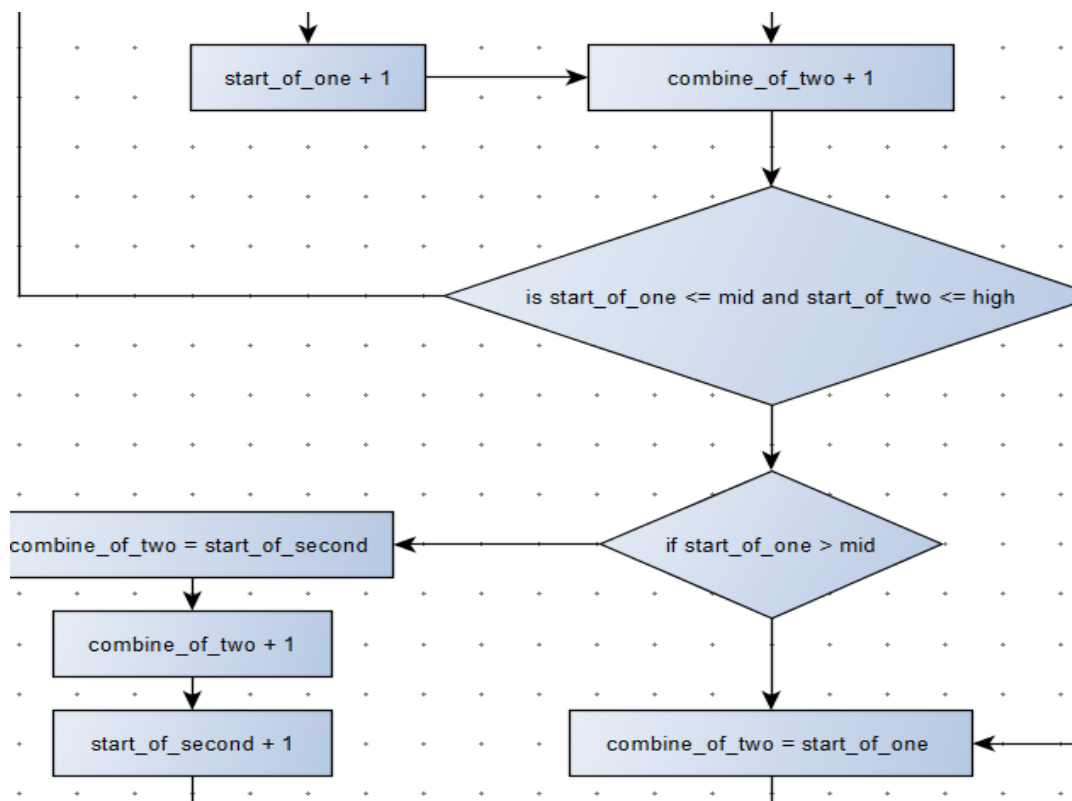
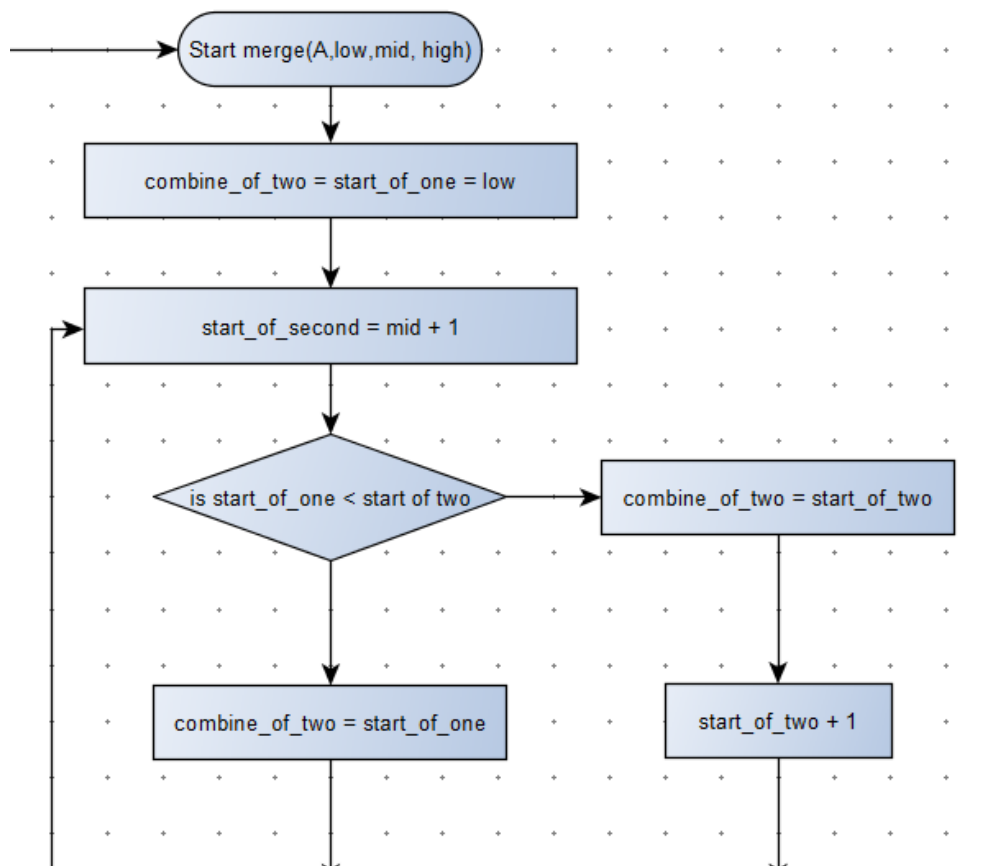


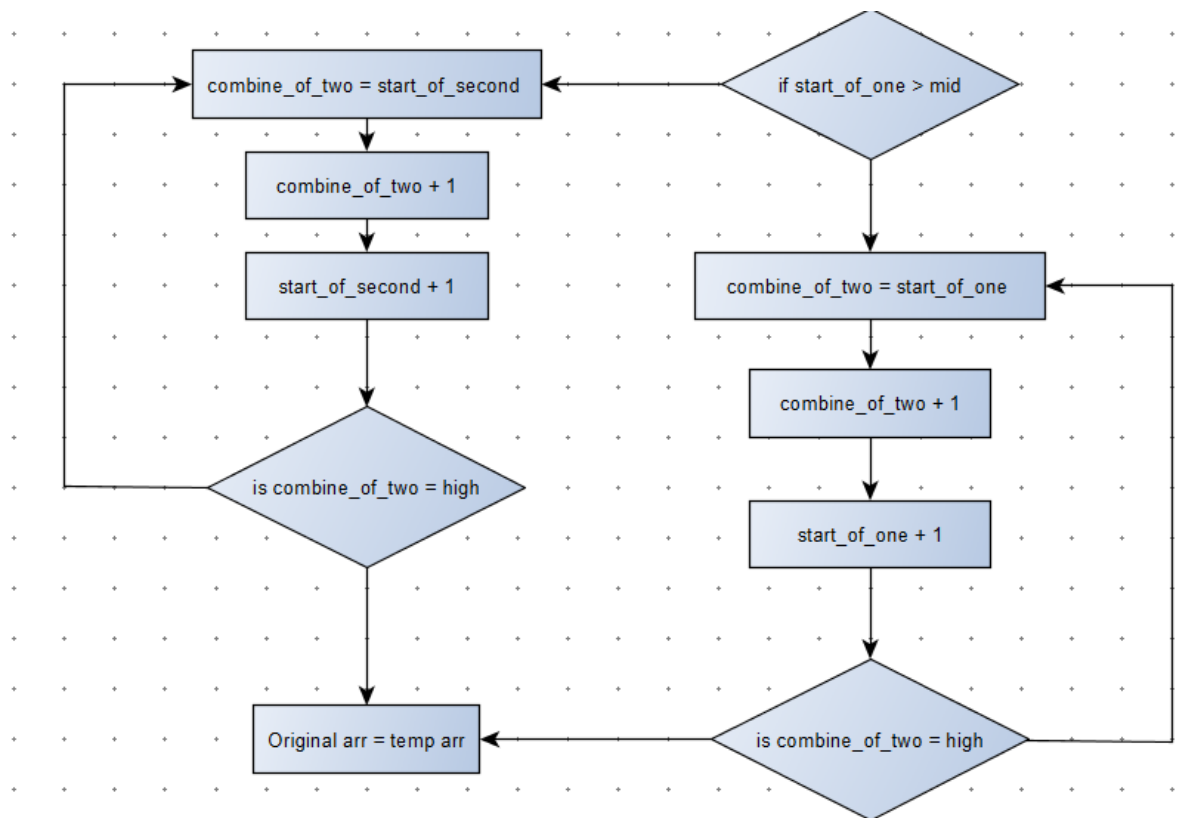




The “temp arr = merge\_sort arr” points to the merge\_sort algorithm







## Question 2 pseudo-code on searching for international students

```
START PROGRAMME

CREATE counter, file pointer

PROMPT for file and point file pointer to it

IF file does not exist, tell the user

ELSE
    WHILE it is not the end of file
    DO
        IF it is international student
        THEN copy it into structure array
        print the student details to the screen
        position plus 1
        counter plus 1
    END WHILE

CLOSE file

END PROGRAMME
```

The Big O for this algorithm is  $O(N)$  as the previous algorithm sorts the students based on student ID. So in the case of searching for international, we don't actually know where each student is. We also know it is not sorted in student status, so we would have no idea where each international is.

There is one loop happening, which is why it has a Big O of N



### Question 3 pseudo-code on searching for a particular student by surname

```
START PROGRAMME

CREATE counter, file pointer

PROMPT for file and point file pointer to it

IF file does not exist, tell the user

ELSE
    WHILE it is not the end of file
    DO
        IF it is the person we want
        print the student details to the screen
        end while loop
    ELSE
        position plus 1
        counter plus 1
    END WHILE

CLOSE file

END PROGRAMME
```

The big O is  $O(N)$  as again we don't have a hundred percent certain the location of the student by surname, so we must check each student linear. There is one loop to check all the students surname.