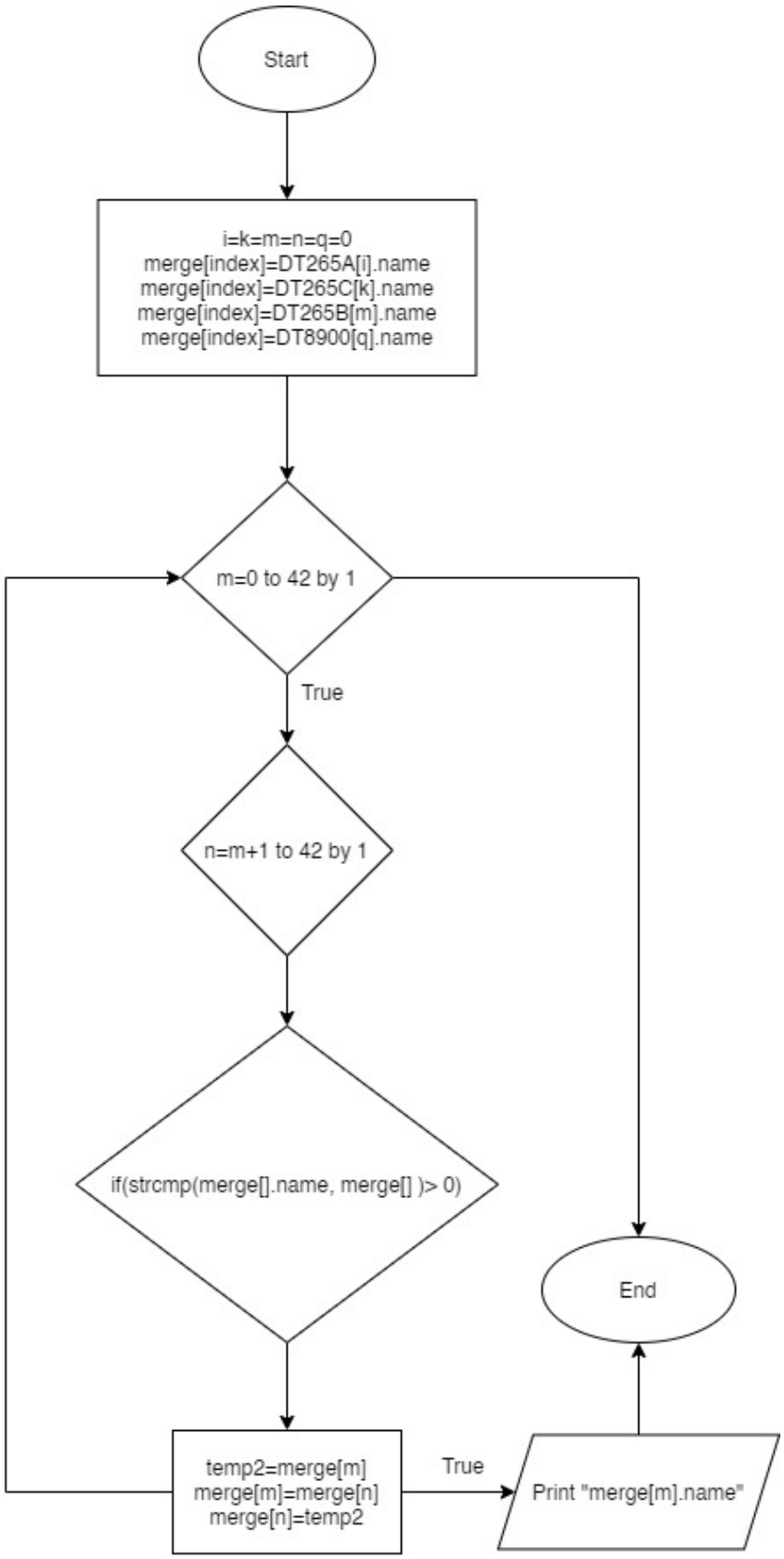
AYAN ABEDIN
D1915792

Algorithm and Design and Problem CMPU1001

Assignment Report

Part I

Flowchart to merge list into one:



I have used ***a simple sorting method*** to combine the four list into a single list.

The algorithm used to sort out the *main list* by surname is **Bubble Sort Algorithm**. Bubble Sort works by repeatedly swapping the adjacent elements if they are in wrong order.

There are 42 names in the combined list, so the algorithm must loop through the program 42 times to arrange all the names followed by the first letter of the names.

The general time complexity of Bubble Sort is **O(n^2**). The Big O for the sorting algorithm is **O (1)**. This is because only a single additional memory space is required i.e., for temp variable. Here the whole combined list is stored in a temp variable. It cannot be **O(n**), as that would be the best-case scenario and the list is not sorted alphabetically. Also, worst case i.e., **O(n^2)** is not applicable since there is no reverse order. Therefore, the Big O is **O (1)**.

Part II

```
Psuedo code

Begin

Function linear(struct[], int)

for i = 0 to (n - 1) by 1 do

if (struct[i] = item) then

Return i

Return -1

Exit

endif

endfor


Int main()

Struct full_list[];

Int x = 1;

Int n = sizeof(struct) / sizeof(struct[0]);
```

```
Int result = search(struct, n ,x);

If(result == -1)

Print

Else

Print "Sorted list is: "

End
```

To search for the full-time students from the list I have used ***Linear Search***. It checks each element of the list sequentially until a match is found or the whole list has been searched.

The time required to search for an element using a linear search algorithm depends on the size of the list. The best-case scenario is when the required element is present at the start of the list and worst case is when present at the last of the list. The Big O of this method is **O(n)**, n is the number of elements.

Part III

Pseudo Code for Searching by Surname

I have used String compare for the search as the running time is ***Olog(n).***

```
START

Char search_name[50];

Merge[m].name; //contains the full sorted list

Int flag, m, n=0;

Print"Student name"

Scanf(%s, &search_name);


For m=0 to m < 42 by 1

If(strcmp(merge[m]).name, search_name == 0)

Print "Student found"

Flag = 0

End if
```

```
End for

If flag ==0

Print "Wrong Input"

End if


END
```

Part IV

The solution to Part IV is in the text file (Algo_assignment.txt).