ECE325 Assignment 1 Report

Team 17

Members: Kleanthi Maria Flouri
Polyvios Charalambous
Lefteris Adamou

The contents of this submission were done exclusively by the members of our team.

References: Virtual Studio Code, GitHub Copilot, YesChat.ai

Part 1

Common elements of sorted lists:

```
Read list1 from user until '#' or max size while ignoring spaces
Save max size of list1 as ‘m’
Read list2 from user until '#' or max size while ignoring spaces
Save max size of list2 as ‘n’
WHILE i < m and j < n
        IF list1[i] > list1[i-1] OR list2[j] > list2[j-1]
                Print “The list was not sorted”
                Exit Program
WHILE index1 < m and index2 < n
        IF list1[index1] < list2[index2]:
                index1+1
                number of comparisons +1
        ELSE IF list1[index1] > list2[index2]:
                index2+1
                number of comparisons +1
        ELSE
                add list1[index1] to common[indexCommon]
                index1+1
                index2+1
                indexCommon+1
                number of comparisons +1
IF there are no common elements
        Display ”No common elements”
ELSE
        Display size of lists, number of comparisons,  list of common elements, sum of
common elements
Exit Program
```

TEST 1:

```
This program finds the common elements between two sorted lists of integers.

Please insert the first sorted integer list. Enter '#' to end the list.
2
3
4
4
6
10
18
#
Please insert the second sorted integer list. Enter '#' to end the list.
2
4
4
10
13
17
#

The size of list 1 is: 7
The size of list 2 is 6
The number of comparisons made is: 8
The common elements are:
2 4 4 10
The sum of the common elements is: 20
```

Part 2

a) Average value of all elements in array

```
Average(array[10]):
        sum = 0
        FOR each element in array:
                sum += element
        RETURN sum / 10
```

b) Difference between the largest and the smallest number

```
Diff_Between_Max_Min(array[10]):
        max = minimum number for integers
        min = maximum number for integers
        FOR each element in array:
                IF element > max → max = element
                IF element < min → min = element
        RETURN max - min
```

c) Finds the index largest number in the array

```
max = minimum number for integers
FOR each element in array:
        IF element > max → max = element
        Maxindex → current_index
END loop
Return Maxindex
```

d) Finds all unique elements

```
unique( array[10], unique[10], size):
```

```
FOR each element in array:
        Flag_unique→TRUE
        FOR each element until j=size
            IF unique = array
                Flag_unique→FALSE
                Exit_loop
            IF Flag_unique→TRUE
                unique=array
                Size++
```

e) Raise value of each element to the power of 2

```
squared_function(array[10])
        FOR each element of array
            array = array*array
```

f) Sort the array

```
FOR each element of array except first (assume first element is sorted)
        index= array
        FOR j=i,  j>0,   j--
            IF index >= left number (compare index_element with its left element
                Exit second loop because its sorted
             ELSE
                Swap elements : array[j]=array[j-1]
array[j]= index (new index)
```

TEST 2:

```
Please enter 10 integers:
40
5
61
3
8
12
45
32
9
23
The average is: 23.80
The difference between the maximum and minimum is: 58
The index of the maximum element is: 3
The unique elements are: 40 5 61 3 8 12 45 32 9 23
The squared elements are: 1600 25 3721 9 64 144 2025 1024 81 529
The sorted elements are:
3 5 8 9 12 23 32 40 45 61
```

Part 3

COMMENTS:

- The students marks range from 0 to 10
- To update a participant record (3) the user must enter the student ID
- When updating a participant record (3) all the information must be added again.

Structure Student:

ID[20], Name[50], participations, project1, project2, total

DECLARE Student students[MAX_STUDENS]

1. Add participant record

```
IF students is full
ELSE
    Read ID
    Read Name
    Read participations
    Read project1
    Read project2
    Calculate total score (0.6*project1 + 0.4*project2)
    Increase studentCount
```

2. Delete participant record

```
IF students is empty (studentCount = 0)
    Print “No participants to delete”
ELSE
    Request ID of participant to delete
    Read ID
    FOR i < studentCount, i++
```

```
            IF students[i] ID = ID that was read
                Replace students[i] with students[i+1] until end of list
                studentCount - 1
```

3. Update Participant Record

```
Request ID of participant to update
    Read ID
    FOR i < studentCount
        IF students[i] ID = Read ID
            Read ID
            Read Name
            Read participations
            Read project1
            Read project2
            Calculate total score (0.6*project1 + 0.4*project2)
            i++
```

4. Save all participant records to a .txt file

```
IF file is not found
    Print "Error"
Open file "participants.txt"
FOR i < studentCount
    Print in file: students ID, Name, participations, project1, project2, total
    i++
Close file
```

5. Load all participant records from a .txt file

```
Open file "participants.txt"
IF file is not found
    Print "Error"
```

Exit function
WHILE a Student students, with the correct format is written in the file
studentCount ++
IF studentCount > maximum students allowed
BREAK from loop
Close file

6. View all participant records

IF student count=0
Print “No students”
Exit function
ELSE
FOR each student
Print “ID”, “Name”, “participation”, “Grade1”, “Grade2”, “Total”

7. View participants by number of previous participations

IF student count=0
Print “No students”
Exit function
ELSE
User types number of participations
FOR each element of array
IF user_input=student.participation
FLAG=TRUE
Print “ID”, “Name”, “participation”, “Grade1”, “Grade2”, “Total” (for matching students participations)
IF FLAG=FALSE
Print “No students found”

8. View participant with highest total score

```
IF student count=0
        Print “No students”
        Exit function
DECLARE maxIndex = 0
FOR i < studentCount
        IF students[i] total > students[maxIndex] total
                maxIndex = i
Print “Participant with the highest score is students[maxIndex]”
```

9. Calculate average total score over all participants

```
IF student count=0
        Print “No students”
        Exit function
sum = 0
FOR each element in array until student count
                sum += student.total
Print sum / student count
```

10. Exit program

```
IF user input = 10
End program
```

TEST 3:

NOTE: The menu appears every time after an operation ends.

```
===============Menu===============
1. Add participant record
2. Delete participant record
3. Update participant record
4. Save all participant records in a .txt file
5. Load all participant records from a .txt file
6. View all participant records
7. View participants by number of previous participations
8. View participant with highest total score
9. Calculate average total score over all participants
10. Exit program
==================================
Enter your choice: 1
Enter ID: 123
Enter Name: student A
Enter Number of previous participations: 2
Enter Project 1 score: 6
Enter Project 2 score: 7
Participant added successfully.

===============Menu===============
1. Add participant record
2. Delete participant record
3. Update participant record
4. Save all participant records in a .txt file
5. Load all participant records from a .txt file
6. View all participant records
7. View participants by number of previous participations
8. View participant with highest total score
9. Calculate average total score over all participants
10. Exit program
==================================
Enter your choice: 6
ID: 123, Name: student A, Participations: 2, Project 1: 6.00, Project 2: 7.00, Total: 6.40
```

```
==================================
Enter your choice: 1
Enter ID: 234
Enter Name: student B
Enter Number of previous participations: 4
Enter Project 1 score: 4
Enter Project 2 score: 9
Participant added successfully.
```

```
==================================
Enter your choice: 6
ID: 123, Name: student A, Participations: 2, Project 1: 6.00, Project 2: 7.00, Total: 6.40
ID: 234, Name: student B, Participations: 4, Project 1: 4.00, Project 2: 9.00, Total: 6.00
```

```
==================================
Enter your choice: 2
Enter ID of participant to delete: 123
Participant deleted successfully.
```

```
==================================
Enter your choice: 6
ID: 234, Name: student B, Participations: 4, Project 1: 4.00, Project 2: 9.00, Total: 6.00
```

```
==================================
Enter your choice: 3
Enter ID of participant to update: 234
Enter new Name: student B updated
Enter new Number of previous participations: 4
Enter new Project 1 score: 5
Enter new Project 2 score: 7
Participant updated successfully.
```

```
==================================
Enter your choice: 6
ID: 234, Name: student B updated, Participations: 4, Project 1: 5.00, Project 2: 7.00, Total: 5.80
```

```
==================================
Enter your choice: 4
DEBUG: saveToFile() entered with 1 students
Records saved successfully.
```

```
==================================
Enter your choice: 1
Enter ID: 345
Enter Name: student C
Enter Number of previous participations: 2
Enter Project 1 score: 7
Enter Project 2 score: 8
Participant added successfully.
```

```
==================================
Enter your choice: 7
Enter number of previous participations: 2
ID: 345, Name: student C, Project 1: 7.00, Project 2: 8.00, Total: 7.40
```

```
==================================
Enter your choice: 7
Enter number of previous participations: 1
No participants found with 1 participations.
```

```
==================================
Enter your choice: 8

Participant with Highest Score:
ID: 345, Name: student C, Participations: 2, Project 1: 7.00, Project 2: 8.00, Total: 7.40
```

```
==================================
Enter your choice: 9
Average total score: 6.60
```

```
==================================
Enter your choice: 4
DEBUG: saveToFile() entered with 2 students
Records saved successfully.
```

```
==================================
Enter your choice: 10
Exiting program.
```

```
===============Menu===============
1. Add participant record
2. Delete participant record
3. Update participant record
4. Save all participant records in a .txt file
5. Load all participant records from a .txt file
6. View all participant records
7. View participants by number of previous participations
8. View participant with highest total score
9. Calculate average total score over all participants
10. Exit program
==================================
Enter your choice: 5
Records loaded successfully.

===============Menu===============
1. Add participant record
2. Delete participant record
3. Update participant record
4. Save all participant records in a .txt file
5. Load all participant records from a .txt file
6. View all participant records
7. View participants by number of previous participations
8. View participant with highest total score
9. Calculate average total score over all participants
10. Exit program
==================================
Enter your choice: 6
ID: 234, Name: student B updated, Participations: 4, Project 1: 5.00, Project 2: 7.00, Total: 5.80
ID: 345, Name: student C, Participations: 2, Project 1: 7.00, Project 2: 8.00, Total: 7.40
```