MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG140 C Programming – Programming Assignment 4

Date handed out: **26 May, 2023, Friday**
Date submission due: **09 June, 2023, Friday, 23:55 (Cyprus Time)**

Programming Assignment 4: Signature Interpreter

Purpose:

The main purpose of this programming assignment is to revise the topics that we have covered in CNG140 including fundamentals of C programming, conditional statements, repetitive statements, functions, arrays, pointers, dynamic memory allocation, files, strings, etc. This is a comprehensive assignment that covers all the topics you have seen in CNG140.

Description:

A signature is the way a person signs her/his signature. The collection of the signature using a digital tablet is represented as a set of data packets tracking pen movements in time. Your task is to implement a C program that can be used to analyze and interpret processed signature data collected using a digital tablet.

In this programming assignment, you will write a program that reads signature records from a file and allows the user to process these records.

As usual, don't try to compile your entire program in one "big bang". Compile it piece by piece. Test each piece that you have compiled to make sure it works correctly before you add the next piece.

Programming Requirements

You will read signature records from a file that includes a table of records shown in Table 1.

- full_name: name and surname of the signature holder separated with a space. Assume always there will be just one name and one surname for each signature.
- x_initial: Initial x-coordinate of the signature.
- x_final: final x-coordinate of the signature.
- y_initial: Initial y-coordinate of the signature.
- y_final: final y-coordinate of the signature.

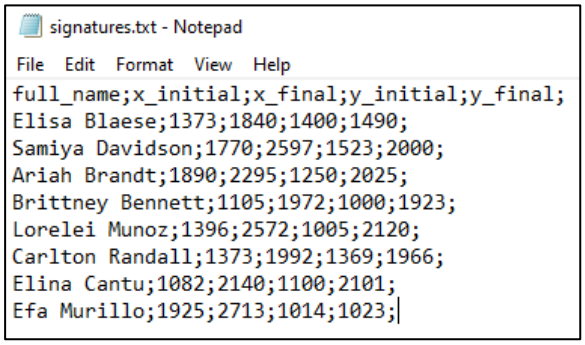
Table 1: Signature records

full_name	x_initial	x_final	y_initial	y_final
Elisa Blaese	1373	1840	1400	1490
Samiya Davidson	1770	2597	1523	2000
Ariah Brandt	1890	2295	1250	2025
Brittney Bennett	1105	1972	1000	1923
Lorelei Munoz	1396	2572	1005	2120
Carlton Randall	1373	1992	1369	1966
Elina Cantu	1082	2140	1100	2101
Efa Murillo	1925	2713	1014	1023

Define a **structure** type to represent one row of this signature record table. The structure will include a string to represent the full_name, and a real number to represent x_initial, x_final, y_initial, y_final, width, and height of the signature.

Write a program to implement the following steps to process such signature records as follows. Before you start, copy the data in Table 1 into a file called "signatures.txt". Please make sure

that the column headings are included and columns are separated with a ; (semicolon) character including the end of the line. For example, the "signatures.txt" should look as follows;



a. Read the name of this data file from the **command line** which is called "signatures.txt". Then check the file, if the file does not exist, your program needs to ask the user for entering the correct file name.

b. Load "signatures.txt" into an array of structures called SignaturesTable. Here, you also need to compute the width and height of the signature (see below-given formulas) and included that into the array you created for each structure/record.
   height= y_final - y_initial
   width= x_final - x_initial

c. Define and call the following functions.

   Load_SignaturesTable – Takes as a parameter the name of the input file. The function creates an array of SignatursTable, opens the given file, fills the SignaturesTable array, closes the file, and returns the SignaturesTable array. Please note that this function needs to return both the actual array as the function result and also the total number of signatures read from the file. Please note that here you cannot return both the table structure and also the size of the array, therefore, you need to return the actual table and use pass-by-reference to update the total number of signatures in the Main function. Please note that you cannot make an assumption about the number of rows in the given table. When you load the data from the file, you also need to compute and add the width and height to the SignaturesTable array.

   Display_SignaturesTable – Takes as parameters the SignaturesTable array and its actual size. Then, displays the table on the screen. If the table has not been loaded then an appropriate error message should be displayed to the user.

   Search – Takes as parameters the SignaturesTable array, its actual size, and a string representing the name and surname of the student. If the entered name and surname is found then this function returns the position of this record in the array otherwise it returns -1 if that signature is not found (details of the searched signature will be displayed in the main function according to the returned value from this search function). Please note that when you search if the given name/surname is part of a name and surname in the array then it will be accepted as found. For example, if the array has "Ahmet Mehmet" then when you search "Ahmet Mehmet", then this function will return found. Similarly, when you search, "Ahmet" or "Mehmet", it will also return true. However, when you search for "met", it should return false. If the list has more than one name then it will return the position of the first signature with the given name.

   Sort – Takes as parameters the SignaturesTable array and its actual size. It then asks the user to enter a sorting option and depending on the input from the user reorders the records of the signatures. The user can sort the data based on the width and height. **Hint**: You learned how to use Bubble Sort (Worksheet 14), you can use that algorithm or another sorting

algorithm. However, clearly comment out in your code which algorithm you use and how you do the sorting.

A sample run will be as follows:

```
>Interpreter sign.txt
This file does not exist. Please enter again: signature.txt
This file does not exist. Please enter again: signatures.txt
The signature records file (signatures.txt) has been successfully loaded!
The following records have been loaded:

  full_name         x_initial   x_final   y_initial   y_final   width  height
  Elisa Blaese      1373        1840      1400        1490      467    90
  Samiya Davidson   1770        2597      1523        2000      827    477
  Ariah Brandt      1890        2295      1250        2025      405    775
  Brittney Bennett  1105        1972      1000        1923      867    923
  Lorelei Munoz     1396        2572      1005        2120      1176   1115
  Carlton Randall   1373        1992      1369        1966      619    597
  Elina Cantu       1082        2140      1100        2101      1058   1001
  Efa Murillo       1925        2713      1014        1023      788    9


To search for a signature, please press 1
To sort signatures based on width or height, please press 2
To exit, please press 3
Your choice:1

Enter the name of the signature owner: Ahmet Mehmet
That owner does not exist! Please try again!
Enter the name of the signature owner: Elina Cantu

Elina Cantu start signing at x=1082 and y=1100 and finalized at x=2140 and
y=2101. Hence, Elina Cantu has a signature of width 1058 and height 1001.

To search for a signature, please press 1
To sort signatures based on width or height, please press 2
To exit, please press 3
Your choice:2

Sort by (w: width, h: height): w

  full_name         x_initial   x_final   y_initial   y_final   width  height
  Lorelei Munoz     1396        2572      1005        2120      1176   1115
  Elina Cantu       1082        2140      1100        2101      1058   1001
  Brittney Bennett  1105        1972      1000        1923      867    923
  Samiya Davidson   1770        2597      1523        2000      827    477
  Efa Murillo       1925        2713      1014        1023      788    9
  Carlton Randall   1373        1992      1369        1966      619    597
  Elisa Blaese      1373        1840      1400        1490      467    90
  Ariah Brandt      1890        2295      1250        2025      405    775


To search for a signature, please press 1
To sort signatures based on width or height, please press 2
To exit, please press 3
Your choice:2
```

```
Sort by (w: width, h: height): h

  full_name         x_initial   x_final   y_initial   y_final   width   height
  Lorelei Munoz     1396        2572      1005        2120      1176    1115
  Elina Cantu       1082        2140      1100        2101      1058    1001
  Brittney Bennett  1105        1972      1000        1923      867     923
  Ariah Brandt      1890        2295      1250        2025      405     775
  Carlton Randall   1373        1992      1369        1966      619     597
  Samiya Davidson   1770        2597      1523        2000      827     477
  Elisa Blaese      1373        1840      1400        1490      467     90
  Efa Murillo       1925        2713      1014        1023      788     9


To search for a signature, please press 1
To sort signatures based on width or height, please press 2
To exit, please press 3
Your choice:3
```

Grading:
If your code does not compile, you will automatically get zero. If your code compiles, you will then be graded based on the following scheme:

Grading Point	**Mark (100)**
Load_SignaturesTable function that reads the file and dynamically populates the data from the file and the computed width and height to the array. You should not make any assumption about the size of the data.	40 points
Display_ SignaturesTable function that displays the array to the user with appropriate messages.	10 points
Search function that searches and looks up for a specific string in the array and display appropriate messages.	20 points
Sort function that sorts the data in the array by the value given by the user.	20 point
Main function that coordinates these functions and extra functions needed.	10 point

Please note that the functions prototype descriptions are provided in Programming Requirements part

Rules:

Please make sure that you follow the restrictions for the assignment as follows:

- Strictly obey the input-output format. Do not print extra things.
- **You are not allowed to use global variables.**
- **You are not allowed to use goto statements.**
- Add your name/surname and ID at the top of your code as comments and name your source file "Name-Surname-StudentID.c"
- Submit your solution as a C file to odtuclass. Do not compress it (zip, rar, …).

If you fail to obey any of the above rules, you will automatically get zero.