**DSA project**

**-**

**Computer Programming Course**

**By Manta Mădălin-Ștefan**

**ETTI - 423G**

1. Description of the problem in natural language

In this project, I created a data structure for representing my genealogical tree, thinking it would be an original and interesting thing to show. I actually wanted to start from a binary tree structure, but because my family is numerous, I adapted it and you can have up to 10 children. I will also add some photos of my real genealogical tree that I followed, which was made recently by one of my cousins, and as you will see I will follow it thoroughly.

To describe the program, it lets you insert new people by defining their name and their gender, in this way adapting the left/right binary tree structure to male/female respectively mother/father structure. After that, we define the child-parent relationship, by telling both their mother and father. I put these things into the code and not into the console because there are numerous names and in this way we save time and we can also avoid mistakes. Then we can call the numerous function implemented to discover somebody’s details.

I created our principal Person structure, with a name, mother, father children, number of children and gender. Then a function to insert a new person into the family tree, to add their children, to find both their descendants and ascendants (and I also used recursive functions), to show us siblings, cousins, grandchildren and their number, and also uncles and aunts.

Last but not least, in the main function I listed my whole family tree, starting from my great grandparents, with their gender and their parents. I used the defined functions multiple times to find different details about some of the members. You can change the names in the function and find everything about whoever you want. I hope you’ll like it!

1. Work flow for the program

A white paper with writing on it

Description automatically generated

1. The listing for the entire program (with useful comments)
2. Instances of running the program (screenshots)

5. Indication of the original parts of the program as well as those taken over, in a bibliographic note (with authors, publication, other relevant data), showing all sources of inspiration (including links to used sites)

Laboratory 9 – TREES.

Course 8, 9, 10 – about trees.

Laboratory 5 – the use of “this” pointer.

<https://www.freecodecamp.org/news/all-you-need-to-know-about-tree-data-structures-bceacb85490c/>

<https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/>

<https://www.codeshikhi.com/2022/01/object-pointers-object-reference-in-oop.html>

<http://www.java2s.com/Code/C/Data-Structure-Algorithm/Basicsofafamilytree.htm>

<https://cboard.cprogramming.com/c-programming/181713-family-tree.html>

<https://c.happycodings.com/beginners-lab-assignments/c-programming-code-basics-of-a-family-tree.html>

<https://stackoverflow.com/questions/49655746/family-tree-in-c>

<https://www.chegg.com/homework-help/questions-and-answers/write-complete-c-program-create-family-tree-using-linked-list-implementation-traverse-tree-q66366238>