# BLG233E

DATA STRUCTURES

CRN:11146

3.HOMEWORK

STUDENTS:

HÜSEYİN ERDOĞAN

040100054

PROJECT DEADLİNE: 25.12.2012

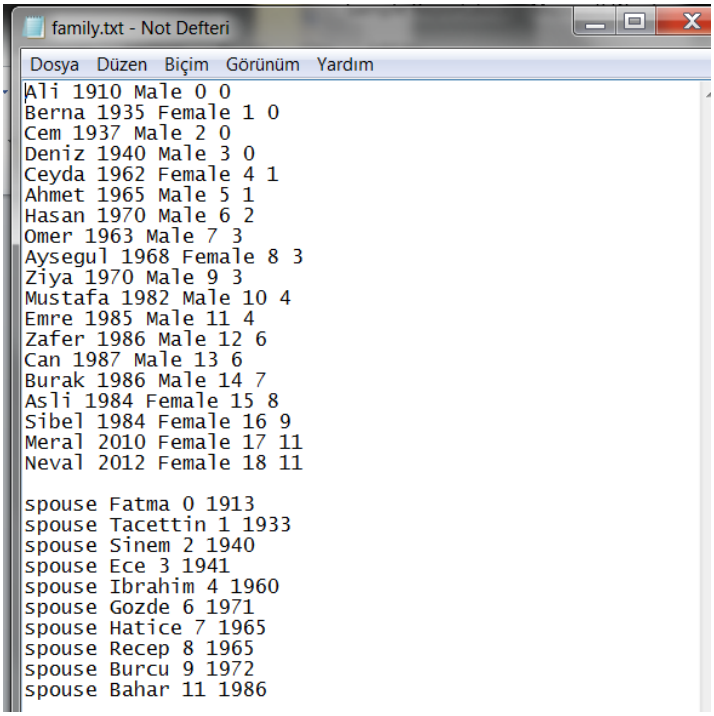
# INTRODUCTION

In this homework, a family will be represented using a tree. Each individual will be represented as a node in the tree, and relationships will be represented with the connections between the nodes. There are two types of people, namely, individual and spouse.

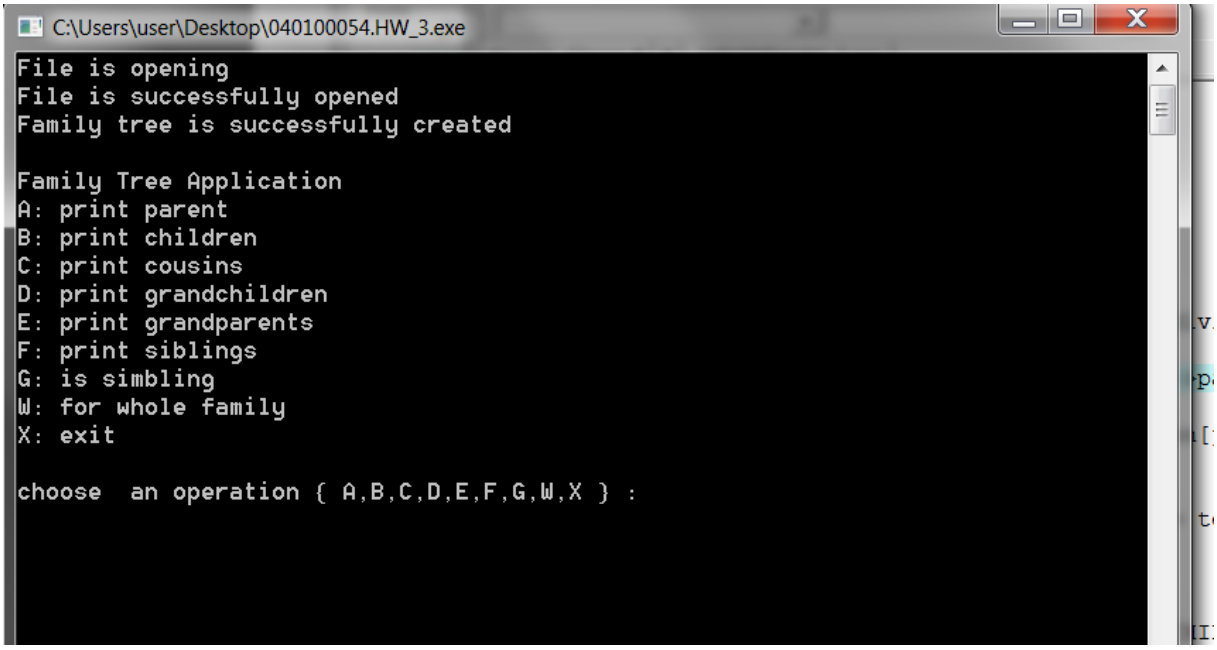
- Each individual have a **name** (char*), **year of birth** (int), **gender** (char), **pointer array to children** (node**) , **pointer to wife/husband** (spouse*), **node ID** (int), and **parent ID** (int).
- The spouse type have two data types: **name** (char*) and **year of birth** (int).

# Development and Runtime environment

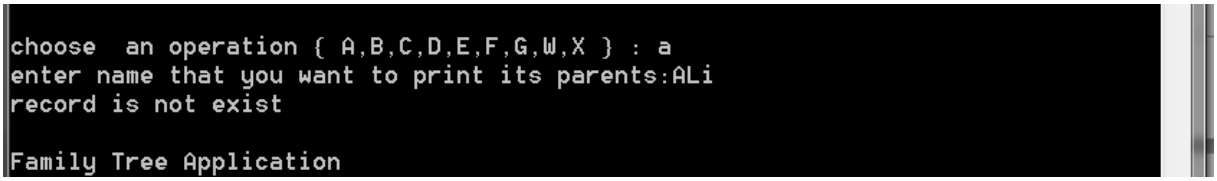
Project is developed in Microsoft Windows7 and Linux Ubuntu 11.10. Geany, g++ and Dev-C++ are used for compiling. Program runs in **main()** function.



When program is started, program gives information about opening file and if there is no problem, program shows family tree menu.

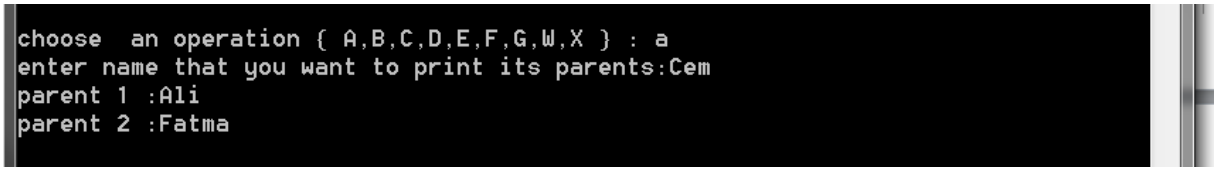



If user enters invalid name, program give an error message.




If user select A option, program wants name that user want to print its parents.

And program print its parents.




If user select B option, program wants name that user want to print its children.

And program print its children.

```
choose  an operation { A,B,C,D,E,F,G,W,X } : b
enter name that you want to print its children:Cem
children 1 :Hasan
```

If user select C option, program wants name that user want to print its cousins.

And program print its cousins

```
choose  an operation { A,B,C,D,E,F,G,W,X } : c
enter name that you want to print its cousins:Ziya
cousin :Ceyda
cousin :Ahmet
cousin :Hasan
```

If user select D option, program wants name that user want to print its grandchildren.

And program print its grandchildren

```
choose  an operation { A,B,C,D,E,F,G,W,X } : d
enter name that you want to print its grandchildren:Ali
grandchildren :Ceyda
grandchildren :Ahmet
grandchildren :Hasan
grandchildren :Omer
grandchildren :Aysegul
grandchildren :Ziya
```

. If user select E option, program wants name that user want to print its grandparents.

And program print its grandparents

```
choose  an operation { A,B,C,D,E,F,G,W,X } : e
enter name that you want to print its grandparents:Ziya
grandparent 1:Ali
grandparent 2:Fatma
```

. If user select F option, program wants name that user want to print its siblings.

And program print its siblings

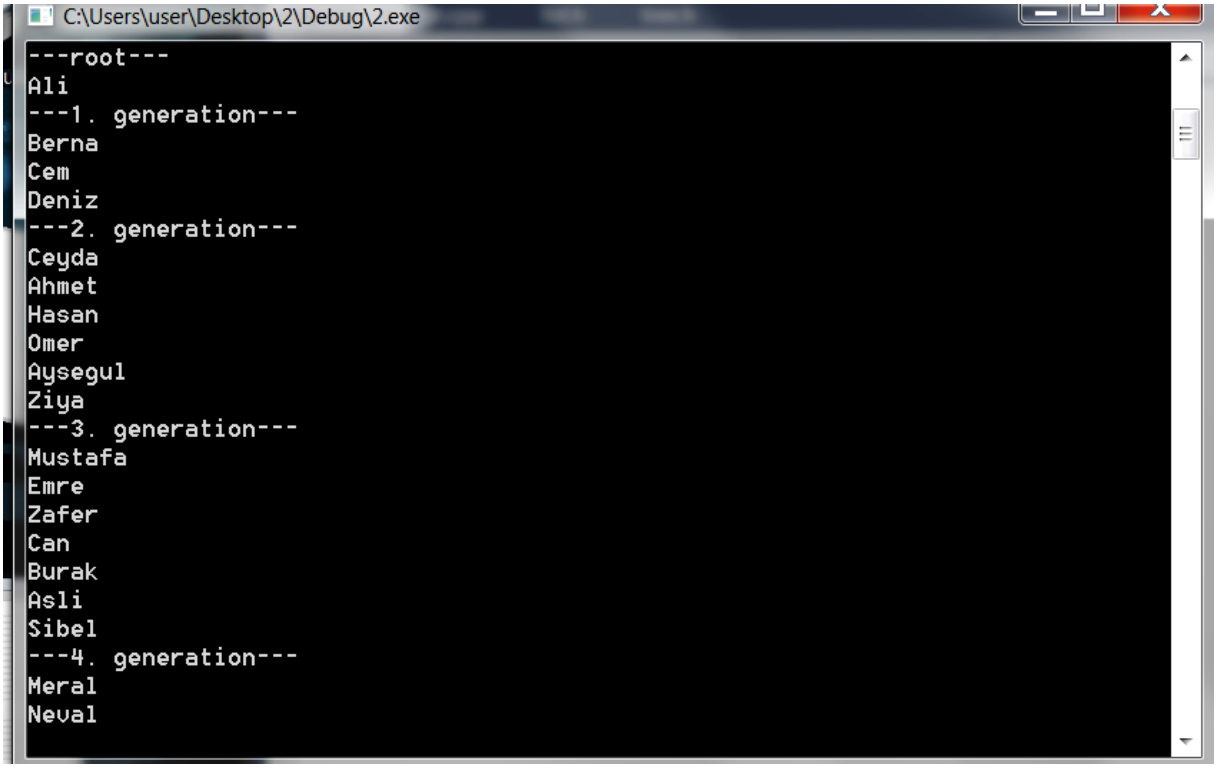
```
choose  an operation { A,B,C,D,E,F,G,W,X } : f
enter name that you want to print its siblings:Ziya
siblings :Omer
siblings :Aysegul
```

. If user select G option, program wants names that user want to they are siblings or not.

And program print information.

```
choose  an operation { A,B,C,D,E,F,G,W,X } : g
enter first name that you want to compare:Ziya
enter second name that you want to compare:Aysegul
Persons are siblings
```

. If user select W option, program print whole tree layer by layer.



# Data Structures and Variables

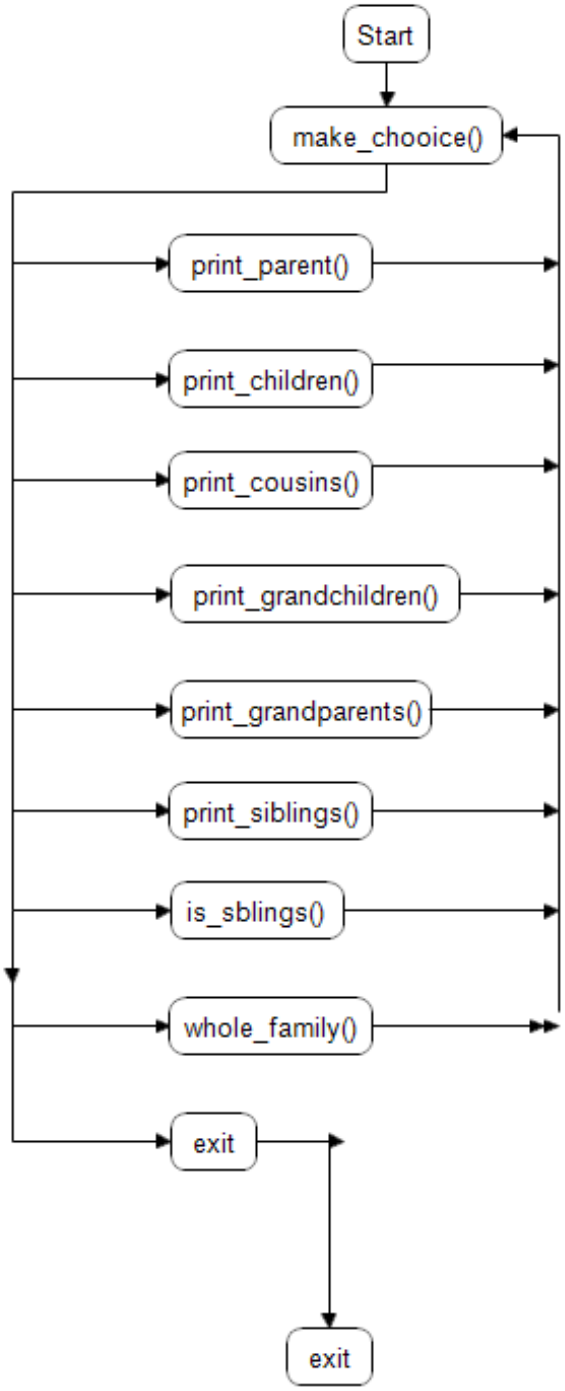
Program runs in **main** function. In addition:

- void print_menu();
- bool make_chooice(char)
- void read_fromfile();
- void print_parent();
- void print_children();
- void print_cousins();
- void print_grandchildren();
- void print_grandparents();
- void print_siblings();
- void is_simbling();
- void whole_family();

And program have list data type in it:

- individual *root;
- void create();
- void empty();
- void Preorder_isearch(individual *,individual *,bool);
- void add_s(spouse *,int);
- void add(individual *);
- void add_spouse(individual *,spouse *,int,bool);
- void make_empty(individual *);
- individual* search_node(char *,individual *,individual *);
- individual* search_parent(individual *,individual *,individual*);
- individual* search_grantparent(char *,individual *);
- void print(individual *);

# Program Flowchart



# Result

In this homework, functions, tree struck and three functions are used . Reading from file , writing to file , adding to tree, finding place to node from tree, finding node's parents in tree, finding node's grandparents in tree, finding node's cousins in tree, finding node's childrens in tree, finding node's grand childrens in tree, finding node's siblings in tree and printing whole tree layer by layer are learned.