

BLG233E

DATA STRUCTURES

CRN:11146

3.HOMEWORK

STUDENTS:

HÜSEYİN ERDOĞAN

040100054

PROJECT DEADLINE: 25.12.2012

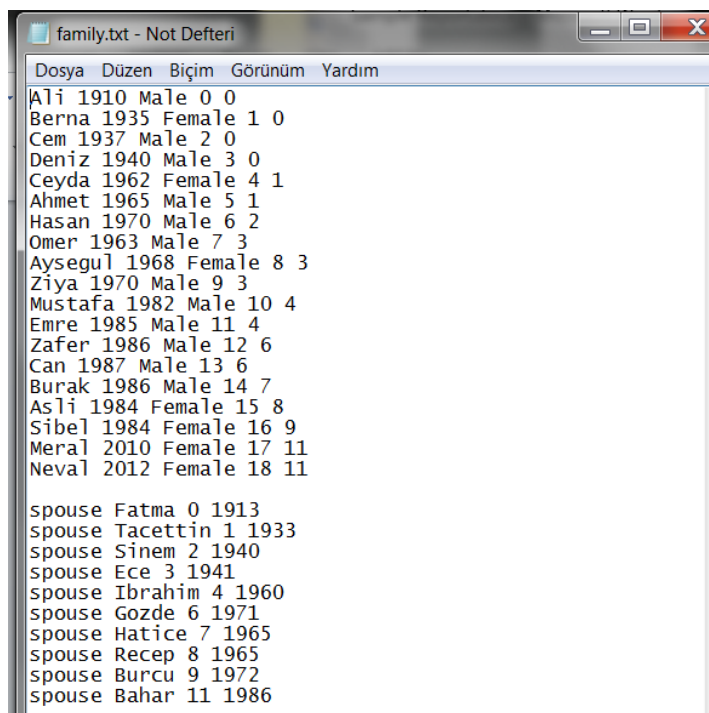
INTRODUCTION

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

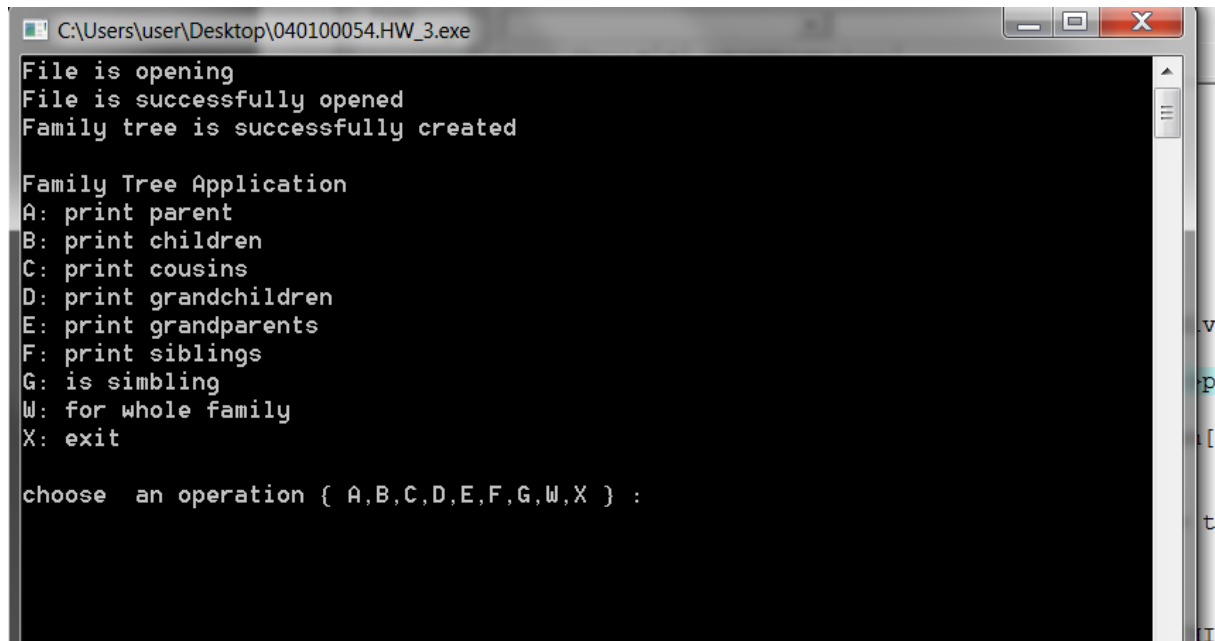
Project is developed in Microsoft Windows7 and Linux Ubuntu 11.10. Geany, g++ and Dev-C++ are used for compiling. Program runs in **main()** function.



```
family.txt - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
Ali 1910 Male 0 0
Berna 1935 Female 1 0
Cem 1937 Male 2 0
Deniz 1940 Male 3 0
Ceyda 1962 Female 4 1
Ahmet 1965 Male 5 1
Hasan 1970 Male 6 2
Omer 1963 Male 7 3
Aysegul 1968 Female 8 3
Ziya 1970 Male 9 3
Mustafa 1982 Male 10 4
Emre 1985 Male 11 4
Zafer 1986 Male 12 6
Can 1987 Male 13 6
Burak 1986 Male 14 7
Asli 1984 Female 15 8
Sibel 1984 Female 16 9
Meral 2010 Female 17 11
Neval 2012 Female 18 11

spouse Fatma 0 1913
spouse Tacettin 1 1933
spouse Sinem 2 1940
spouse Ece 3 1941
spouse Ibrahim 4 1960
spouse Gozde 6 1971
spouse Hatice 7 1965
spouse Recep 8 1965
spouse Burcu 9 1972
spouse Bahar 11 1986
```

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

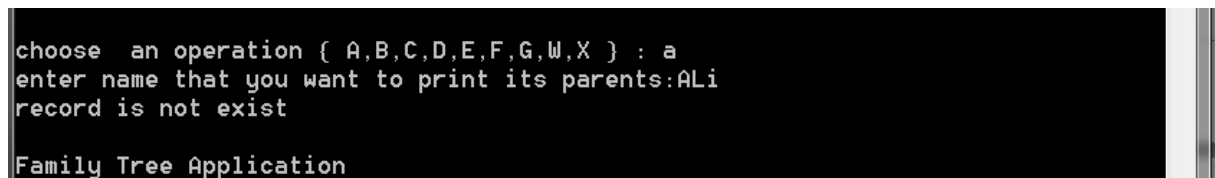


```
C:\Users\user\Desktop\040100054.HW_3.exe
File is opening
File is successfully opened
Family tree is successfully created

Family Tree Application
A: print parent
B: print children
C: print cousins
D: print grandchildren
E: print grandparents
F: print siblings
G: is simbling
W: for whole family
X: exit

choose an operation { A,B,C,D,E,F,G,W,X } :
```

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

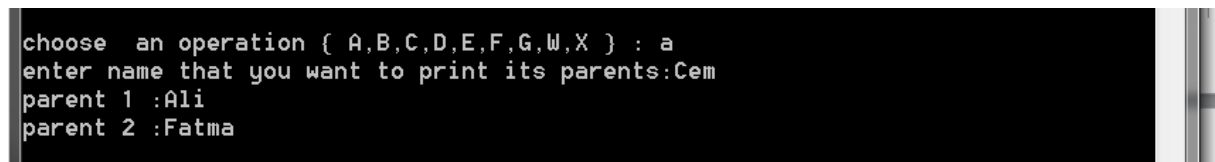


```
choose an operation { A,B,C,D,E,F,G,W,X } : a
enter name that you want to print its parents:ALi
record is not exist

Family Tree Application
```

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

And program print its parents.



```
choose an operation { A,B,C,D,E,F,G,W,X } : a
enter name that you want to print its parents:Cem
parent 1 :Ali
parent 2 :Fatma
```

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.

```
C:\Users\user\Desktop\2\Debug\2.exe
---root---
Ali
---1. generation---
Berna
Cem
Deniz
---2. generation---
Ceyda
Ahmet
Hasan
Omer
Aysegul
Ziya
---3. generation---
Mustafa
Emre
Zafer
Can
Burak
Asli
Sibel
---4. generation---
Meral
Neval
```

Data Structures and Variables

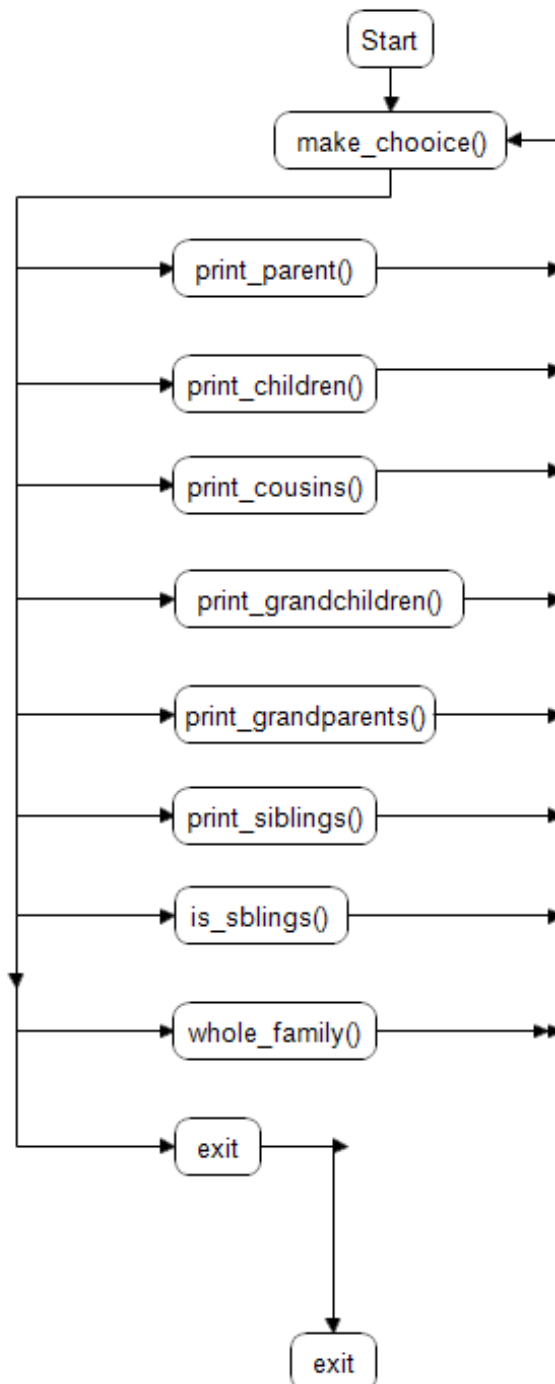
Program runs in **main** function. In addition:

- 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 struct 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.