

PONDER 13 : GENEALOGY

Due Mon. at 11:59 PM MST

The purpose of this assignment is to combine the knowledge you have accumulated this semester to solve a problem.   We have an old LDS church GEDCOM genealogical file that we would like to display the individual ancestors first in alphabetical order by last name, then by first name, then by birthday.  Then we would like to display the ancestors of “Greg Cameron” from his genealogical tree by generation.  The genealogical tree can be built from the information in the GEDCOM file.  This requires a level order traversal of the tree to display each generation.

When designing the solution to this problem you must use your own data structures.  You may not use any data structures from the Standard Template Library.

Program Requirements:

Your program must get the GEDCOM filename from the command line. Find a copy of this file in /home/cs235/week13\_E/cameron.ged.    Parse the file in order to find each individual, their given name, last name, birthdate and individual id number (this will be needed later in the program).   As an individual is extracted from the GEDCOM file, insert the individual in alphabetical order into a linked list. The list should be sorted by last name, then by given name, then by birth date (if necessary).  Write the list of sorted individuals to a file called sorted.dat.  Compare “sorted.dat” with the file “final1.out”.  The files must match EXACTLY.  Use the command: (diff sorted.dat final1.out) to compare them. The file should contain one line per person and should look as follows:

Gregory Lawrence Cameron, b. 8 Nov 1963

Lawrence Clyde Cameron, b. 22 Dec 1941

Paul Roland Cameron, b. 2 Jul 1908

. . .

After all the individuals have been inserted into a linked list and written to the file, the program should build a pedigree tree structure of all the individuals.   Set the mother and father pointers for each person in the list to point to the correct father and mother.  Use the family records that follow the individual records in the file to find out this information.  Once the tree is built, display this pedigree to the screen  using  a level order traversal  starting with “ Gregory Lawrence Cameron” as the root of the tree.  Display this pedigree to the screen.  See the file “final2.out” to see the expected output format.  This "pedigree" should be printed to the screen in the following manner:

The Ancestors of Gregory Lawrence Cameron:

Parents:

Lawrence Clyde Cameron b. 22 Dec 941

Bonnie Lorraine Downen b. 28 Mar 1945

Grand Parents:

Paul Roland Cameron b. 2 Jul 1908

Margaret Ferne Simdars b. 7 Dec 1911

Edward Rodney Downen b. 10 Nov 1917

Mildred Claire Marsh b. 10 Dec 1919

Great Grandparents

. . .

Compare the output of your program to the screen with the file “final2.out”.  This can be easily done with the following Linux command:

a.out cameron.ged | diff - final2.out

or

a.out cameron.ged > generations.dat

diff generations.dat final2.out

The output of your program must match the file final2.out **exactly!**

YOU MAY NOT USE THE STL CONTAINERS, ITERATORS OR ALGORITHMS ON THIS ASSIGNMENT!!!

The files cameron.ged, final1.out, and final2.out can be found in the directory /home/cs235/week13\_E

**TestBed**

There is not test bed for this program.  Your "sorted.dat" file must match the file  "final1.out".  
Your screen output of the pedigree must match the file "final2.out". Compare your output using the Linux diff utility.

Submitting

You will submit this assignment individually using the Linux submit command.

1. Create your own makefile which compiles the program and builds the TAR file.
2. Submit your file using the submit command.

submit week13.tar