COMPUTER SCIENCE 21A (SPRING TERM, 2024)
DATA STRUCTURES AND ALGORITHMS

PROGRAMMING ASSIGNMENT 0

Tudor Kings and Queens of England

You are to write a program that uses recursion to find all family ancestors and descendants of a given person.

The program asks for the name of the family data file and then prompts for a specific person's name. If the person is in the family database, the program shows all the ancestors and all the descendants for that person. For ancestors it must show all parents, all grandparents, all great grandparents, etc. For descendants it must show all children, all grandchildren, all great grandchildren, etc. The program also must use indentation to make it clear who is a parent of whom and who is a child of whom. As a test data file, you will be using information about the Tudor kings and queens of England. The name of the file is "tudor.txt".
Format of `tudor.txt`: The first section of the file (until the line with "END" has all the unique family members. The section of the `tudor.txt` after the "END" is divided into groups of three - first the family member, then his/her mother, then his/her father. This is how you can tell who is related to whom.

i.e.:

`Arthur` –> Family Member
`Elizabeth of York` -> Mother
`Henry VII` -> Father
`Henry VIII` –> Family Member
`Elizabeth of York` -> Mother
`Henry VII` -> Father

etc.

Special cases: The data file contains no information prior to Henry VII, so when you ask for information about him, you get a lot of descendants but no ancestor information. Below is the log of execution you should obtain for him.

```
What is the input file? tudor.txt
Person's name ('quit' to end)? Henry VII

Ancestors:
    Henry VII
Descendants:
    Henry VII
        Arthur
        Henry VIII
            Mary I
            Elizabeth I
            Edward VI
        Margaret
            James V
                Mary, Queen of Scots
                    James VI & I
            Margaret Stuart
                Henry, Lord Darnley
                    James VI & I
        Mary
            Frances
                Lady Jane Grey
```

At the other extreme is James VI of Scotland who became James I of England. The data file has no information about his descendants, but a great deal of information about his ancestors. Below is the log you should obtain for him.

```
What is the input file? tudor.txt
Person's name ('quit' to end)? James VI & I
Ancestors:
    James VI & I
        Mary, Queen of Scots
            Mary of Guise
            James V
                Margaret
                    Elizabeth of York
                    Henry VII
                James IV
        Henry, Lord Darnley
            Margaret Stuart
                Margaret
                    Elizabeth of York
                    Henry VII
Descendants:
    James VI & I
```

Your task: You are required to demonstrate that your program works for these two individuals and for another in the middle of this family tree. In particular, you should test it for Margaret. Below is the log you should obtain for her.

```
What is the input file? tudor.txt
Person's name ('quit' to end)? Margaret
Ancestors:
    Margaret
        Elizabeth of York
        Henry VII
Descendants:
    Margaret
        James V
            Mary, Queen of Scots
                James VI & I
        Margaret Stuart
            Henry, Lord Darnley
                James VI & I
```

Your client program should be named "**`Relatives.java`**" and it should include only static methods. It should prompt the use for the test file and then for the name of the person. If the name does not exist it should print a message and quit. If the name exists it should print the ancestors and descendants as shown in the above examples.

Design Hints:

Feel free to structure your program according to your preferences but remember that a crucial aspect of being a proficient software engineer is paying careful attention to the design of your program.

Consider structuring your program to enable the client program to read the provided family file, store family information, and utilize **recursive methods** to print ancestors and descendants when given a person's name.

Your grade will reflect the quality of your object-oriented design skills.

Make sure that you're writing object-oriented code and **that the ancestors and descendants are produced using recursive methods**. You should also explain these methods in your README.txt file.