MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG242 Programming Language Concepts – Assignment 1 - Haskell

Date Handed Out: 31 March 2023
Submission Due: 14 April 2023 – 23:59

Please Read This Page Carefully

Submission Rules:

- **You should provide your name, surname, and ID, as well as a comment on the first line of your files**, stating that you read the rules specified here and the submission is your own work. **Submissions without this statement will not be graded.** An example statement can be provided as follows:

```
-- Zekican Budin – 1234567
-- I read and accept the submission rules and the extra rules specified
in each question. This is my own work that is done by myself only.
```

- Please refer to the syllabus[1] provided for CNG 242 for the measures in place in case of any academic dishonesty[2,3].
- The instructors or TAs may ask for demo sessions for any of the submissions.
- You cannot share this worksheet with any third parties. Upon doing so, any detected action will directly be sent to the disciplinary committee.
- You need to submit a single .hs file **named with your student id** only. For example, **1234567.hs . Any other files or attachments will not be graded.**
- You should read the questions fully and follow the directions listed in there**.**
- You can only get full marks if your file fully compiles, runs, and generates correct output for all possible cases and inputs.
- You should only submit one solution per question. Your solution might have multiple lines or include self-written helper functions. However, only the functions with the same name will be graded.
- The assignment should not be shared publicly in any manner, at any time. The assignment cannot be disclosed or disseminated to anyone before, during, or after the submission.

[1] Page 3&4 (Course rules, #1,2,3)
[2] Taking unfair advantage in assessment is considered a serious offence by the university, which will take action against any student who contravenes the regulation through negligence or deliberate intent.
[3] For a comprehensive cheating definition, please refer to: https://ncc.metu.edu.tr/res/academic-code-of-ethics . When a breach of the code of ethics occurs (cheating, plagiarism, deception, etc.), the student will be added to the BLACKLIST.

Assignment I: Family Trees

Learning Outcomes: On successful completion of this assignment, a student will:

- Have practiced how to program using functional programming languages;
- Have used different approaches to define data types in Haskell;
- Have practiced how to use recursive definitions in functional programming paradigm

In this assignment, we are going to work with tree structures. The case study we are going to consider is a "Family Tree". Various free and practical software tools exist to store and analyse information related to family trees, such as https://familytreemagazine.com/websites/best-free-genealogy-websites/ or https://www.familyecho.com/

For simplicity we will **only consider family members of same gender and with no inbreeding** (the tree is established from either all male or all female) [1]. Family trees are commonly used to represent genealogical charts. Advanced and detailed family trees are used effectively in medicine and social work and are mostly known as genograms. Vertices of the family trees are used to represent the members of a family and edges are used to represent the parent–child relationships. The terminology used for the analysis of trees is mostly adopted from disciplines such as botany and genealogy.

Consider a rooted tree *T*. Let *v* be a vertex in *T* other than the root. The unique vertex *u* where there is a directed edge from *u* to *v is called as the* **parent** of *v*. Vertices with the same parent are called **siblings**. If a vertex of a rooted tree has no children, it is called a **leaf** and all the other vertices are called as **internal vertices**.

Part I: Data Type Definition [5 points]

In this part of the assignment, you should introduce a "FamilyTree" data type to fulfil all requirements to represent a family tree shown in Part I of the assignment.

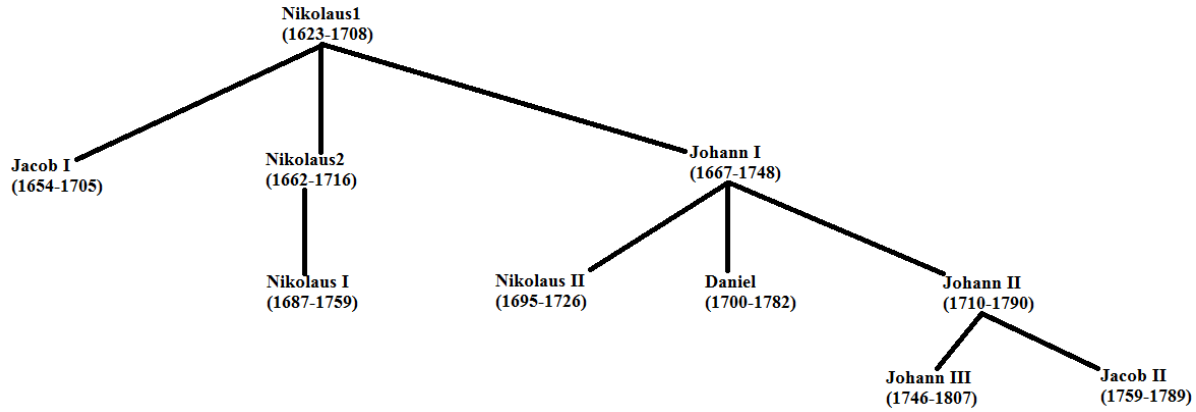

Figure 1 The Bernoulli Family of Mathematicians

Part II: Family Tree [50 points]

In this assignment, given a list of unique names, list of integer values for the number of children (one gender only; the tree is established from either all male or all female) for each person and a list of 2-tuples (pairs) for birth and death years of each person, your program will establish the family tree. Please consider the family tree of the male members of the Bernoulli family of Swiss mathematicians in Figure 1 as an example. Figure 1 is used as an example for the following sample run;

Function name: familytree
Input: list of names, list of number of children, and years lived
Output: The family tree
Sample run:
familytree [“Nikolaus1”, “Jacob I”, “Nikolaus2”, “Nikolaus I”, “Johann I”, “Nikolaus II”, “Daniel”, “Johann II”, “Johann III”, “Jacob II”] [3, 0, 1, 0, 3, 0, 0, 2, 0, 0] [(1623, 1708), (1654, 1705), (1662, 1716), (1687,1759), (1667, 1748), (1695, 1726), (1700, 1782), (1710, 1790), (1746, 1807), (1759, 1789)]

Part III: Functions [45 points]
In this part of the assignment, you will also introduce some functions to analyse the tree* as follows:

Function name: search **[10 points]**
Description: This function will return True if the given person is in the given Tree otherwise it should return False. The return type should be Boolean, do not return a string for this question.

Input:	tree, name of the person
Output:	True or False (Boolean)
Sample run:	search familytree “Daniel” True search familytree “Austin” False

Function name: parent **[10 points]**
Description: This function will return the name of the parent for a specific person (message “No Parent” if person is the root, if the person does not exist, message “Not Found”)

Input:	tree, name of the person
Output:	name of the parent
Sample run:	parent familytree “Daniel” “Johann I” parent familytree “Michael” “Not Found”

Function name: siblings **[10 points]**
Description: This function will return the names of the siblings for a specific person as a list (message [] if person has none or person does not exist.)

Input:	tree, name of the person
Output:	name of the sibling
Sample run:	siblings familytree "Johann III" ["Jacob II"]

Function name: avglifetime **[15 points]**
Description: This function will return the average lifetime of the family.

Input:	tree
Output:	average lifetime
Sample run:	avglifetime familytree 62.7

*** All of the outputs in Part III are generated based on the given figure in Part I.**
**** You may use existing questions as helpers to your functions.**

Additional Rules:

- Strictly obey the specifications, input output formats. Do not print extra things.
- Solutions without a FamilyTree data type will not be accepted.
- Code quality, modularity, readability will be the part of grading.
- Your variable names and comments must be written in English.

Assessment Criteria

The assignment will be marked as follows:

Item	Marks (Total 100)
Implementation of Family Tree	50
Formulation data type	5
search function	10
parent function	10
siblings function	10
avglifetime function	15

References

1. Rosen, Kenneth. Discrete Mathematics and Its Applications 7th edition. McGraw-Hill, 2011.
2. Wenger, Rephael. Isosurfaces: Geometry, Topology, & Algorithms. CRC Press, 2013.