

## CSI2372 – Fall 2021

### Assignment 1

Due Date: 5<sup>th</sup> October

#### Question 1 of 2) **Life Predictor**

The Social Security Administration maintains an actuarial life table that contains the probability that a person in the United States will die (<http://www.ssa.gov/OACT/STATS/table4c6.html>). The death probabilities for 2009 are stored in the file `LifeDeathProbability.txt` which is available in the Assignment 1 files. In this file, there are three values for each row: the age, death probability for a male, and death probability for a female. For example, the first five lines are:

```
0  0.006990  0.005728
1  0.000447  0.000373
2  0.000301  0.000241
3  0.000233  0.000186
4  0.000177  0.000150
```

The interpretation for the fourth line is that a 3-year-old female has a 0.000186 chance of dying during.

Write a program that inputs an age and sex from the keyboard in the main function. The main function should call a function named `simulate` (that you must write), sending in the age and sex as parameters. The function should simulate to what age a person will live by starting with the death probability for the given age and sex. You can do this by reading the data from the file row by row. Skip rows that are less than the input age. Once the input age is reached, generate a random number between 0-1 and if this number is less than or equal to the corresponding death probability then predict that the person will live to the current age and return that age. If the random number is greater than the death probability then increase the age by one and repeat the calculation for the next row in the file.

If the simulation reaches age 120, then stop and predict that the user will live to 120. The main function should output the simulated age for when the person will die. This program is merely a simulation and will give different results each time it is run, assuming you change the seed for the random number generator.

Question 2 of 2) **High Score Manager**

Write a program that manages a list of up to 10 players and their high scores in the computer's memory (not on disk as files). Use two arrays to manage the list. One array should store the player's name and the other array should store the player's high score. Use the index of the arrays to correlate the name with the score. Do not use a struct or class for this program. Your program should support the following features:

- a. **Add** a new player and score (up to maximum of 10 players).
- b. **Print** all players and their score to the screen
- c. Allow the user to enter a player name and output that player's score or a message if the player's name has not been entered (i.e., **search** a player's name)
- d. Allow the user to enter a player name and **remove** the player from the list

Create a menu system that allows the user to select which option to invoke. The features must be in forms of functions.