EC327 – Introduction to Software Engineering

Lab 5 – Arrays, File I/O, Makefiles

The problem for Lab 5 focuses on the practical utilization of C++ data types, user inputs, conditional branches, loops, functions, header files, makefiles, file IO, and arrays.

You need to accept the GitHub classroom assignment (https://classroom.github.com/a/5N2Ym3vi) and clone the repo to your computer. Make sure that your code compiles with the provided framework and runs as expected before getting checked off.

Mode Computation

ModeMain.cpp, readfile.h, readfile.cpp, mode.h, mode.cpp

In this problem, you need to implement several functions across several .h and .cpp files that read an array of integers from a file, find the mode(s) of the array, and print out the results.

Already provided files to use/alter:

- ModeMain.cpp (No need to change)
  It defines the main() function and printModes(). Make sure to look at this file to understand how your functions will be used!
- Makefile (sample already in the repo, but you WILL need to edit this to match the files used here!)
- Several sample input files: input_01.txt, input_02.txt, input_03.txt, input_04.txt (test cases, do not modify them.)

Four required files to fill out:

- readfile.h (function prototype: readFiletoArray())
- readfile.cpp (function implementation: readFiletoArray())
- mode.h (function prototype: findModes())
- mode.cpp (function implementation: findModes())

Two required functions:

- void readFiletoArray (char filename[], int* data);
  This function reads the data stored in the file defined by filename into an integer array, int data[]. You can assume that the main function initializes the data array with enough space to hold the given test data.
- void findModes (int input[], int size, int& frequency, int result[], int& result_count);
  This function takes in the inputs, finds the mode(s), and stores the results in the result array. It also stores the number of modes in result_count, and the frequency of the modes in frequency.

Examples:

Input: 2, 3, 4, 4, 3, 1, 2, 1, 2, 3
Modes: 2, 3
Frequency : 3

Input: 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
Modes: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Frequency : 1

Input: 1, -3, 4, -9, -3, -8, 5, 0, -3, -3
Modes: -3
Frequency : 4

Input: 2, 3, 4, 4, 5, 1, 2, 1, 5, 3
Modes: 1, 2, 3, 4, 5
Frequency : 2

Some assumptions you can make:

- Both data and result arrays have enough space to store all the necessary data and mode results.
- There are 10 input numbers in all the input .txt files.
- The input data are integers in the range from -10 to 10 inclusive (this assumption will make finding the mode significantly easier!!!).

Hints and cautions:

- Pointers * declaration and dereferencing character
- & gets the address of a character
- Pass by value vs. pass by reference

```
double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
double *p = balance;
for (int i = 0; i < 5; i++)
    cout << *(p + i) << endl;
```

Output:

```
1000
2
3.4
17
50
```

Make sure to look carefully at the provided main function to understand how your variables are being used and any additional relationships between variables.