CSCI 340 - Operating Systems I

Assignment 1, Total Points 100

Objectives

In this assignment you will develop a small software program using the C programming language. This assignment will allow you to gain experience in the following areas:

- **Unix Development**: This includes writing, compiling, executing, and debugging C programs.

- **Basic Programming Concepts**: This includes variable declaration, data types, arrays, pointers, operators, expressions, selection statements, looping statements, functions, and header files.

- **File Input/Output (IO)**: This includes reading data from an ASCII text file, and displaying information to the console.

- **Structures**: This includes using the definition of a *struct* data type.

Provided Files

The four files listed below are provided to you.

- **vectors.txt**: Each line in this text file is a vector. Specifically, a vector is defined by a:

  1. magnitude (r), where $r \in \mathbb{R}$
  2. direction (θ) measured in degrees, where $\theta \in \mathbb{R}$

  The format of the file is very simple, each line contains two decimal numbers separated by a comma. The first value is the vector magnitude, and the second value is the vector direction. **Please note**:

  - If the provided vector direction is > 360, then your code must subtract 360 from θ, i.e. $\theta = \theta - 360$, and
  - If the provided vector direction is < -360, then your code must add 360 to θ, i.e. $\theta = \theta + 360$.

  Lastly, this text file defines 10 vectors, and <u>under no circumstance</u> should it be altered or modified in any way.

- **utils.h**: Header file that defines a vector *struct* used in this assignment, and the function prototypes to be completed by you. **Please note**: You <u>may not</u> add new function definitions to this header file.

- **utils.c**: The file containing the <u>implementation</u> of the functions listed in *utils.h*. Having a different file for the implementation separates interface (the include file) from the implementation (the .c file).

- **hw1.c**: Source code file that includes a stubbed out version of the main function, and defines the libraries and constants used in this assignment. **Please note**: You <u>may not</u> remove, modify, or add (i.e. #include) additional libraries. The ones that are provided, are the only libraries needed for this assignment.

The *hw1.c*, *utils.h* and *utils.c* files contain many comments that provide: 1) basic definitions, 2) specific calculations, 3) variable restrictions, and 4) basic step-by-step instructions. Please read the comments carefully and follow their instructions.

Todo

Using the *utils.h* file, you must provide working implementations within the corresponding *utils.c* file for the following function prototypes:

1. *int read(char* file_name, v_struct* p_vec_array)*

2. *double x_component(v_struct* p_vector)*

3. *double y_component(v_struct* p_vector)*

For each function prototype listed above, numerous comments are provided in the header file to guide you in this assignment. Please read them carefully, they either provide basic step-by-step instructions, or basic calculations.

In the *hw1.c* file, you must fully implement the main function, i.e., *int main(int argc, char** argv)*. Please read the comments carefully, they provide the steps to be coded by you in the main method, and how the vector information is to be displayed in the console.

Collaboration and Plagiarism

This is an **individual assignment**, i.e. **no collaboration is permitted**. Plagiarism will not be tolerated. Submitted solutions that are very similar (determined by the instructor) will be given a grade of zero. Please do your own work, and everything will be OK.

Submission

Create a compressed tarball, i.e. *tar.gz*, that only contains the completed *hw1.c*, *utils.h* and *utils.c* files. The name of the compressed tarball must be your last name. For example, *ritchie.tar.gz* would be correct if the original co-developer of UNIX (Dennis Ritchie) submitted the assignment. Only assignments submitted in the correct format will be accepted (no exceptions). Submit the compressed tarball (via OAKS) to the Dropbox setup for this assignment. You may resubmit the compressed tarball as many times as you like, Dropbox will only keep the newest submission.

To be fair to everyone, late assignments will not be accepted. Exceptions will only be made for extenuating circumstances, i.e. death in the family, health related problems, etc. You will be given a week to complete this assignment. Poor time management is not excuse. Please do not email assignment after the due date, it will not be accepted. Please feel free to setup an appointment to discuss the assigned coding problem. I will be more than happy to listen to your approach and make suggestions. However, I cannot tell you how to code the solution. Additionally, code debugging is your job. You may use the debugger (gdb) or print statements to help understand why your solution is not working correctly, your choice.

Grading Rubric

For this assignment the grading rubric is provided in the table shown below.

Program Compiles	10 points
Program Runs with no errors	10 points
read() function implementation	30 points
x_component() function implementation	10 points
y_component() function implementation	10 points
main() function implementation	30 points

In particular, the assigment will be graded as follows, if the submitted solution

- does not compile: 0 of 100 points

- compiles but does not run: 10 of 100 points

- compiles and runs with errors: 15 of 100 points

- compiles and runs without errors: 20 of 100 points

- all functions correctly implemented: 100 of 100 points