

This assignment is meant to get you started with C++ programming. There are two objectives in this assignments.

1. Learn how to write and compile console-based C++ programs
2. Refresh your C programming.

There is only one part in this assignment.

1 Finding integer roots for equations with integer coefficients

The purpose of this exercise is:

1. Refresh your C programming skills. In this exercise you will need to use if-conditionals, loops, basic logic.
2. Efficiency. A key advantage of C/C++ programming is its efficiency. You should try to make your code as efficient as possible. For this exercise, you should carefully examine your code to ensure your code is efficient.

You are given an array of integers A , which represent the coefficients of an equation with a single variable x . These coefficients are ordered from low to high. That is, $A[0]$ is the constant term, $A[1]$ is for x , $A[2]$ is for x^2 and so on. For example, suppose $A = [6, -5, 1]$, the equation is for $6 - 5x + x^2$. You are to implement which finds the smallest non-negative root of the equation within certain range:

```
int ECFindRoot(int *listCoeffs, int degree, int xmax) ;
```

where `listCoeffs` is a pointer to an array of coefficients, `degree` represents the highest degree of the equation, and `xmax` is the maximum value of the root. This function returns the smallest root found (whose value is at most `xmax`) or -1 if there is no root.

The following are what you need to do.

1. You should first ensure your code passes the provided test cases. These test cases are included in the autograder tests. If your code passes these test cases, you earn some points.
2. Then you should ensure your code runs efficiently when the degree of the equation increases. You should conduct thorough analysis of your code with a specific focus on the run-time of the code. Suppose d is the maximum degree of equation. Evaluate the running time for checking for a specific value of x , whether x is the root. If your code for this step runs in $O(d^2)$, it may be considered too slow. You should consider optimizing your code to achieve an efficient run-time of $O(d)$ for this step. We will have run-time check for this.

2 Reverse a singly linked list

This is a popular job interview coding question. You are given a singly linked list. Your goal is reversing it. For example, if a linked list contains three items sequentially: 1 (the head), 2 and 3. Your code should return the new head (3), which points to 2 (which in turn points to 1).

You should reverse the linked list **in place**. That is, you should not need to allocate extra spaces for reversing. Instead, you should just change the pointers in the given list to achieve the reversal. Also, your code should run in time linear to the size of the linked list.

Note: this problem is a common coding problem and you probably can use ChatGPT to complete the task. Don't do that. The purpose of this exercise is getting more familiar with C programming. You need this skill to be able to complete more complex tasks later in this course.

3 Instructions for submission

Please read the following carefully.

1. Submit your programs in GradeScope. A link is provided from HuskyCT. Note that you can submit multiple files by drag and drop into the submission UI.
2. You must use the filenames as specified. If you change the filename, the Autograder won't work.
3. Please don't create your own files. Autograder won't know about the files you created (not in the provided starter code) and won't run test cases properly.
4. For automated grading, we mainly focus on functionalities. However, sometimes we will also test on (i) efficiency, (ii) code complexity, and (iii) code quality. We may use third-party tools for some of these tasks. While these tools have been around for sometime, they can still sometimes produce not so accurate report about your code. If this is the case, contact the TAs about the issues.