

# ECS 140A Homework 01

Sunday, April 10, 2022, 11:59pm PT

## Instruction

- For each of the following problems, solve it using the list of steps described in the syllabus. Otherwise, you will get many points even if your code is completely correct.
- This is a warm-up exercise to get you familiar with all three languages. Each problem is to be written in all of them. You may use whichever you are most comfortable with as the first language. (See syllabus for details).
- **Submission:** You **MUST** submit **one pdf file per question** and **all of the source files compressed in a .zip folder**.
- **Searching and copying codes from online sources is considered cheating.** You may search for syntactical details of the languages. But searching for solution is forbidden and will be reported to SJA.
- You do NOT need to use command line arguments to handle user inputs.
- Don't forget to come up with your own test cases to fully test your code and list your test cases at the end of the document.
- **Do NOT put your name in the submitted files unless you don't mind others knowing who you are when peer reviewing.**

## Problem 1

There are three kinds of parentheses:

- (, )
- {, }
- [, ]

Given a string, write a function called `matching_parentheses` that checks if the parentheses in it match, i.e.,

- open brackets are closed by the same kind of closing brackets; and
- open brackets are closed in the correct order.

Your function should return `true` if the parentheses match, `false` otherwise. The input string may contain any of the parentheses and alphabetical letters. (Return `true` if the input string is empty. You may assume the strings are not too long.)

**Language:** Write the function in **C++, Python, and Rust**. You may use whichever you are most comfortable with as the first language. (See syllabus for details.)

**Note on Rust:**

- There are two kinds of strings in Rust: string literal (`&str`) and string object (`String`). For more details: check

- [https://www.tutorialspoint.com/rust/rust\\_string.htm](https://www.tutorialspoint.com/rust/rust_string.htm)

- <https://stackoverflow.com/questions/24158114/what-are-the-differences-between-rusts-string>

- Based on the description of the problem, `&str` would be more appropriate since the input never changes. But `String` is easier to work with so we will use that.

- Input type: `String`  
Output type: `bool`

#### Examples:

1. Input: `()`  
Output: `true`
2. Input: `[a(b)]`  
Output: `true`
3. Input: `[a(b)]`  
Output: `false`
4. Input: `a{abc([])}`  
Output: `false`
5. Input: `aabc([])}c`  
Output: `false`

## Problem 2

Given an array of strings, we want to find the longest common prefix of the strings. For example, given an array of strings `["apple", "app", "aple", "appl"]`, the longest common prefix is `"ap"`. If the strings do not have any common prefix, return the empty string.

**Language:** Write the function in **C++, Python, and Rust**. You may use whichever you are most comfortable with as the first language. (See syllabus for details.)

- C++:  
Input type: `vector<String>&`  
Output type: `String`
- Rust:  
Input type: `Vec<String>`  
Output type: `String`