# CPH-Leetcode Extension

A Visual Studio Code extension to enhance the competitive programming experience by integrating LeetCode problem test cases directly into your development workflow. This extension allows you to fetch test cases from a LeetCode problem URL and run your code against the fetched inputs to verify its correctness.

---

## Features

### 1. Fetch Test Cases from LeetCode

- Enter a LeetCode problem URL, and the extension fetches input and output test cases for the problem.
- If the test case files already exist, the extension will not recreate them. Instead, it will open the existing files for viewing.
- The fetched or existing test cases are saved into files for easy access:
  - `<problem_name>_inputs.txt`: Contains the input test cases.
  - `<problem_name>_outputs.txt`: Contains the expected output for the test cases.

### Key Functions

1. fileExists
   - This function handles fetching input and output test cases for a given LeetCode problem URL.
2. Run and Compare
   - This function executes the user's code against the fetched test cases and compares the outputs.

### 2. Run and Compare

- Test your solution directly from the editor by running it against the fetched inputs.
- The extension compares your code's output with the expected output.
- Displays detailed information about:
  - Input
  - Expected Output
  - Your Output
  - Test case pass/fail status

### 3. Multi-Language Support

- Supports popular competitive programming languages:
  - C++
  - Python

---

## Installation

1. Clone this repository or download the `.vsix` file.
2. Install the extension:
    - Open Visual Studio Code.
    - Go to **Extensions** (`Ctrl+Shift+X`).
    - Click on the three-dot menu and select **Install from VSIX**.
    - Choose the `.vsix` file to install.
3. Reload Visual Studio Code.

---

## Usage

### Fetch Test Cases

1. Open the Command Palette (`Ctrl+Shift+P` or `Cmd+Shift+P` on macOS).
2. Run the command: `CPH-Leetcode: Fetch Test Cases`.
3. Enter the **LeetCode problem URL** (e.g., `https://leetcode.com/problems/two-sum`).
4. Select your programming language (C++ or Python).
5. The extension creates:
    - A code template file (`<problem_name>.<extension>`).
    - Two files: `<problem_name>_inputs.txt` and `<problem_name>_outputs.txt`.

### Run and Compare

1. Write your solution in the generated file.
2. Open the Command Palette (`Ctrl+Shift+P` or `Cmd+Shift+P` on macOS).
3. Run the command: `CPH-Leetcode: Run and Compare`.
4. The extension will:
    - Execute your code for each input test case.
    - Compare your code's output with the expected output.
    - Display the results in the output channel.

---

## Example Workflow

1. **Fetch Test Cases**:
    - URL: `https://leetcode.com/problems/two-sum`
    - Files generated:
        - `two-sum_inputs.txt`
        - `two-sum_outputs.txt`
        - `two-sum.cpp` (or `two-sum.py`)
2. **Write Code**:
    - Open the generated `two-sum.cpp` (or `two-sum.py`).
    - Write your solution in the provided code template.

3. **Run and Compare**:
    - Run the `Run and Compare` command.
    - See results in the output channel.

---

## Extension Commands

| Command | Description |
| --- | --- |
| `CPH-Leetcode: Fetch Test Cases` | Fetch test cases from a LeetCode URL. |
| `CPH-Leetcode: Run and Compare` | Run your solution and compare outputs. |

---

## Requirements

- **Programming Languages**:
    - For Python: Ensure `python` is installed and available in your system's PATH.
    - For C++: Ensure `g++` is installed and available in your system's PATH.
- **Workspace**:
    - Open a folder in Visual Studio Code before using the extension.

---

## Known Issues

- Currently supports only C++ and Python.
- Ensure internet connectivity to fetch test cases.

---

## Future Enhancements

- Add support for more programming languages like Java, JavaScript, etc.
- Automatically detect language from file extension.
- Allow customization of test case storage paths.

---

## Contributing

Feel free to open issues or contribute to the project by submitting pull requests. Contributions, bug reports, and feature requests are welcome!

---

## Author

Developed with ♥ by [Your Name].