CP2406 Prac-4

By solving the following exercises, you can practice the material discussed in the relevant chapter. Solutions to all exercises are available in the “solutions”-subfolder. However, if you are stuck on an exercise, first reread parts of the chapter to try to find an answer yourself before looking at the solutions.

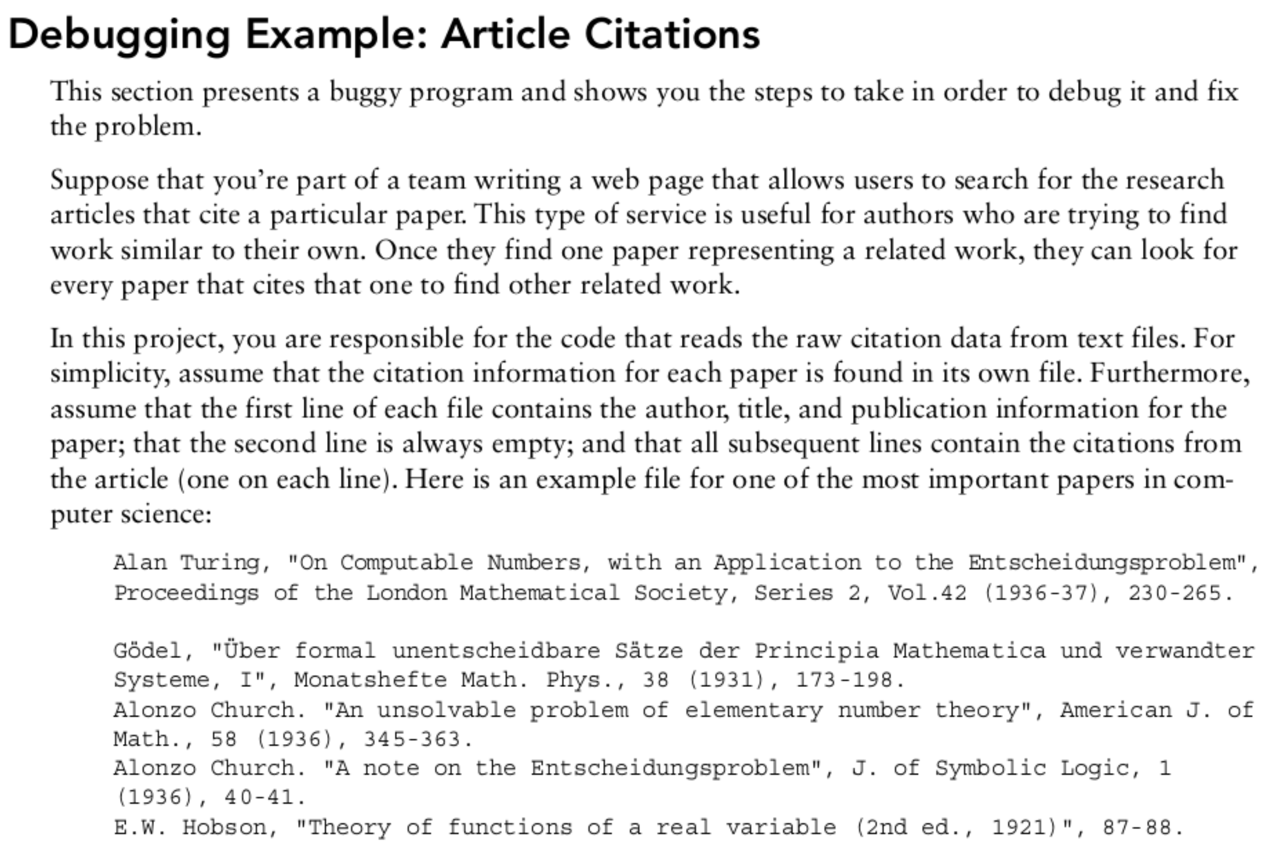
prac04_task01 for Chapter-31-”Debugging”

1) Download and unzip the prac’s code.

2) Open the folder “prac04_tasks” in VSCode.

3) Navigate to “prac04_task01_ArticleCitations” folder.

4) In this task we will work through the Chapter-31 textbook’s debugging example “Article Citations”, page-1067.



Debugging Example: Article Citations

This section presents a buggy program and shows you the steps to take in order to debug it and fix the problem.

Suppose that you’re part of a team writing a web page that allows users to search for the research articles that cite a particular paper. This type of service is useful for authors who are trying to find work similar to their own. Once they find one paper representing a related work, they can look for every paper that cites that one to find other related work.

In this project, you are responsible for the code that reads the raw citation data from text files. For simplicity, assume that the citation information for each paper is found in its own file. Furthermore, assume that the first line of each file contains the author, title, and publication information for the paper; that the second line is always empty; and that all subsequent lines contain the citations from the article (one on each line). Here is an example file for one of the most important papers in computer science:

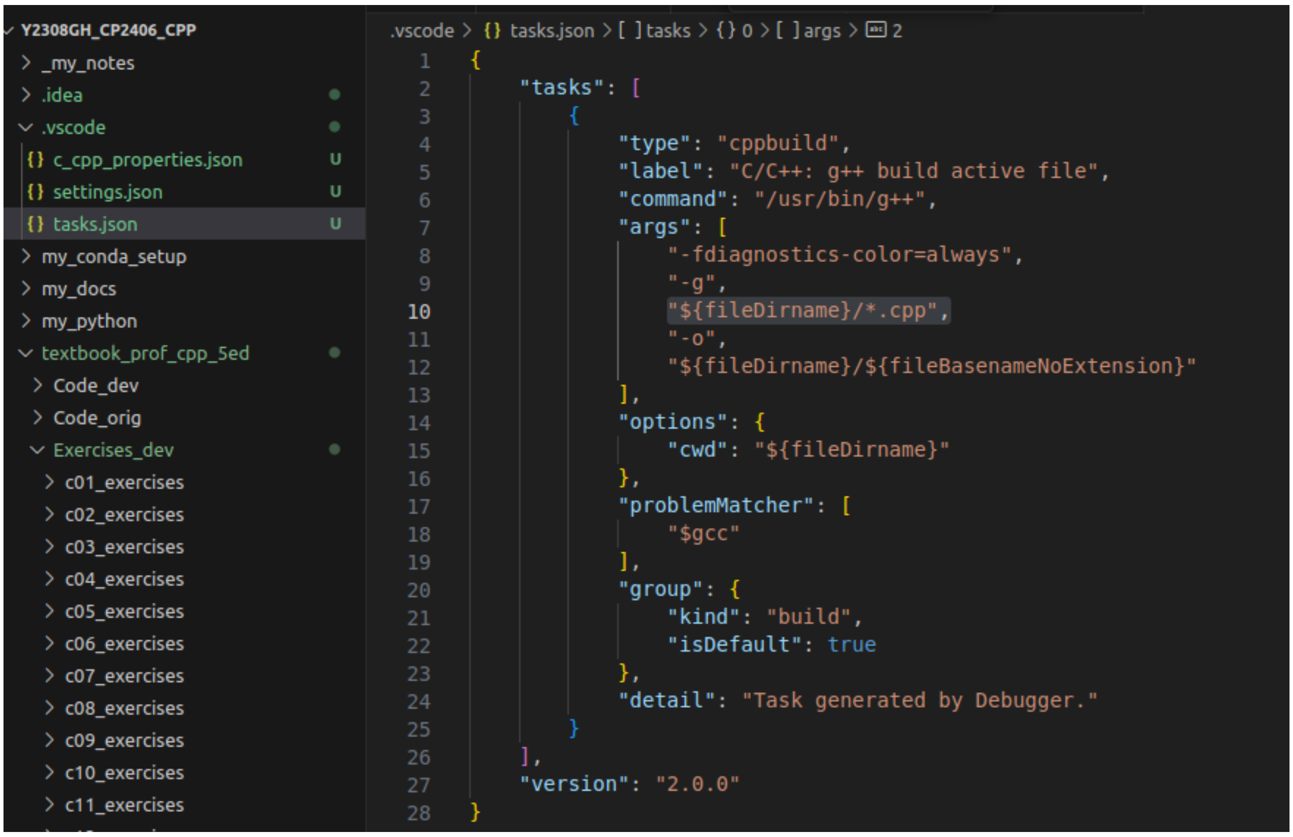
```
Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem",
Proceedings of the London Mathematical Society, Series 2, Vol.42 (1936-37), 230-265.

Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter
Systeme, I", Monatshefte Math. Phys., 38 (1931), 173-198.
Alonzo Church. "An unsolvable problem of elementary number theory", American J. of
Math., 58 (1936), 345-363.
Alonzo Church. "A note on the Entscheidungsproblem", J. of Symbolic Logic, 1
(1936), 40-41.
E.W. Hobson, "Theory of functions of a real variable (2nd ed., 1921)", 87-88.
```

5) Navigate to “01_FirstAttempt_std11” folder.
ADVANCED students navigate to the textbook original “01_FirstAttempt” folder.

6) Let's compile and run "ArticleCitationsTest.cpp".
Do you get errors? This the first time we are trying to compile and run program with multiple source code files.

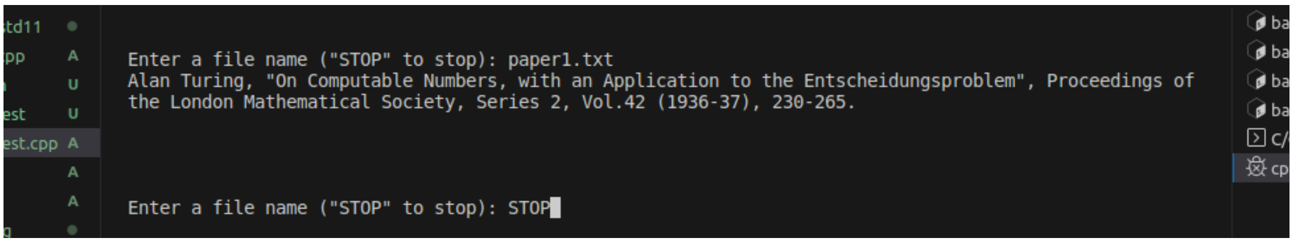
7) We need to configure VSCode to compile **all** .cpp files from the current directory. To do that, open "tasks.json" and type >"${fileDirname}/*.cpp",<



```
{
    "tasks": [
        {
            "type": "cppbuild",
            "label": "C/C++: g++ build active file",
            "command": "/usr/bin/g++",
            "args": [
                "-fdiagnostics-color=always",
                "-g",
                "${fileDirname}/*.cpp",
                "-o",
                "${fileDirname}/${fileBasenameNoExtension}"
            ],
            "options": {
                "cwd": "${fileDirname}"
            },
            "problemMatcher": [
                "$gcc"
            ],
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "detail": "Task generated by Debugger."
        }
    ],
    "version": "2.0.0"
}
```

8) Now you should be able to run "ArticleCitationsTest.cpp" because we included all .cpp files as source code files for compiler and linker.

9) Type: paper1.txt and you should get something like the following:



```
Enter a file name ("STOP" to stop): paper1.txt
Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of
the London Mathematical Society, Series 2, Vol.42 (1936-37), 230-265.



Enter a file name ("STOP" to stop): STOP
```

10) Examine "paper1.txt" and you will see that it has four other citations, which were not printed. This is a bug!

11) Debug "ArticleCitationsTest.cpp" and step through the code in debugger. Locate the problem and but do not fix it yet. Reflect: was it easy to find the bug? Did it take long to find it?

12) Now let's practice message-based debugging, which is also known as "tracing", see page-1071:

Message-Based Debugging

For this bug, you decide to try log-based debugging, and because this is a console application, you decide to just print messages to `cout`. In this case, it makes sense to start by looking at the function that reads the citations from the file. If that doesn't work right, then obviously the object won't have the citations. You can modify `readFile()` as follows:

```
void ArticleCitations::readFile(string_view filename)
{
    // Code omitted for brevity.

    // First count the number of citations.
    cout << "readFile(): counting number of citations" << endl;
    while (!inputFile.eof()) {
        // Skip whitespace before the next entry.
        inputFile >> ws;
        string temp;
        getline(inputFile, temp);
        if (!temp.empty()) {
            cout << "Citation " << count << ": " << temp << endl;
            ++count;
        }
    }

    cout << "Found " << count << " citations" << endl;
    cout << "readFile(): reading citations" << endl;
    if (count != 0) {
        // Allocate an array of strings to store the citations.
        m_citations = new string[count];
        m_numCitations = count;
        // Seek back to the start of the citations.
        inputFile.seekg(citationsStart);
        // Read each citation and store it in the new array.
        for (count = 0; count < m_numCitations; ++count) {
            string temp;
            getline(inputFile, temp);
            if (!temp.empty()) {
                cout << temp << endl;
                m_citations[count] = temp;
            }
        }
    } else {
        m_numCitations = -1;
    }
    cout << "readFile(): finished" << endl;
}
```

13) To speed up your typing, go to "02_CoutDebugging" and copy/paste the "cout<<" tracing messages (see above the bold font lines).

14) You should get something like the following:

Running the same test with this program gives the following output:

```
Enter a file name ("STOP" to stop): paper1.txt
readFile(): counting number of citations
Citation 0: Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und
verwandter Systeme, I", Monatshefte Math. Phys., 38 (1931), 173-198.
Citation 1: Alonzo Church. "An unsolvable problem of elementary number theory",
American J. of Math., 58 (1936), 345-363.
Citation 2: Alonzo Church. "A note on the Entscheidungsproblem", J. of Symbolic
Logic, 1 (1936), 40-41.
Citation 3: E.W. Hobson, "Theory of functions of a real variable (2nd ed.,
1921)", 87-88.
Found 4 citations
readFile(): reading citations
readFile(): finished
Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem",
Proceedings of the London Mathematical Society, Series 2, Vol.42 (1936-37), 230-265.
[ 4 empty lines omitted for brevity ]
Enter a file name ("STOP" to stop): STOP
```

As you can see from the output, the first time the program reads the citations from the file, to count them, it reads them correctly. However, the second time, they are not read correctly; nothing is printed between "readFile(): reading citations" and "readFile(): finished" — why not? One way to delve deeper into this issue is to add some debugging code to check the state of the file stream after each attempt to read a citation:

15) Reflect: Do you feel it was easier to find the bug using tracing? Why? We found the bug's location but not the reason!

16) Let's add more tracing and trace the i/o status, see page-1072:

```
void printStreamState(const istream& inputStream)
{
    if (inputStream.good()) { cout << "stream state is good" << endl; }
    if (inputStream.bad()) { cout << "stream state is bad" << endl; }
    if (inputStream.fail()) { cout << "stream state is fail" << endl; }
    if (inputStream.eof()) { cout << "stream state is eof" << endl; }
}

void ArticleCitations::readFile(string_view filename)
{
    // Code omitted for brevity.

    // First count the number of citations.
    cout << "readFile(): counting number of citations" << endl;
    while (!inputFile.eof()) {
        // Skip whitespace before the next entry.
        inputFile >> ws;
        printStreamState(inputFile);
        string temp;
        getline(inputFile, temp);
        printStreamState(inputFile);
        if (!temp.empty()) {
            cout << "Citation " << count << ": " << temp << endl;
            ++count;
        }
    }
```

```
        cout << "Found " << count << " citations" << endl;
        cout << "readFile(): reading citations" << endl;
        if (count != 0) {
            // Allocate an array of strings to store the citations.
            m_citations = new string[count];
            m_numCitations = count;
            // Seek back to the start of the citations.
            inputFile.seekg(citationsStart);
            // Read each citation and store it in the new array.
            for (count = 0; count < m_numCitations; ++count) {
                string temp;
                getline(inputFile, temp);
                printStreamState(inputFile);
                if (!temp.empty()) {
                    cout << temp << endl;
                    m_citations[count] = temp;
                }
            }
        } else {
            m_numCitations = -1;
        }
        cout << "readFile(): finished" << endl;
    }
```

17) To speed up your typing, go to “03_CoutDebuggingWithStreamState” and copy/paste the “cout<<” tracing messages and printStreamState (see above the bold font lines).

18) Run the program again and you will get:

When you run your program this time, you find some interesting information:

```
Enter a file name ("STOP" to stop): paper1.txt
readFile(): counting number of citations
stream state is good
stream state is good
Citation 0: Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und
verwandter Systeme, I", Monatshefte Math. Phys., 38 (1931), 173-198.
stream state is good
stream state is good
Citation 1: Alonzo Church. "An unsolvable problem of elementary number theory",
American J. of Math., 58 (1936), 345-363.
stream state is good
stream state is good
Citation 2: Alonzo Church. "A note on the Entscheidungsproblem", J. of Symbolic
Logic, 1 (1936), 40-41.
stream state is good
stream state is good
Citation 3: E.W. Hobson, "Theory of functions of a real variable (2nd ed.,
1921)", 87-88.
stream state is eof
stream state is fail
stream state is eof
Found 4 citations
readFile(): reading citations
stream state is fail
stream state is fail
stream state is fail
stream state is fail
readFile(): finished
```

19) Read the following explanation and fix of the bug from the textbook, page 1074:

It looks like the stream state is good until after the final citation is read for the first time. Because the paper1.txt file contains an empty last line, the while loop is executed one more time after having read the last citation. In this last loop, inputFile >> ws reads the whitespace of the last line, which causes the stream state to become eof. Then, the code still tries to read a line using getline(), which causes the stream state to become fail and eof. That is expected. What is not expected is that the stream state remains as fail after all attempts to read the citations a second time. That doesn't appear to make sense at first: the code uses seekg() to seek back to the beginning of the citations before reading them a second time.

Chapter 13 explains that streams maintain their error states until you clear them explicitly; seekg() doesn't clear the fail state automatically. When in an error state, streams fail to read data correctly, which explains why the stream state is also fail after trying to read the citations a second time. A closer look at the code reveals that it fails to call clear() on the istream after reaching the end of the file. If you modify the code by adding a call to clear(), it will read the citations properly.

Here is the corrected readFile() method without the debugging cout statements:

```cpp
void ArticleCitations::readFile(string_view filename)
{
    // Code omitted for brevity.

    if (count != 0) {
        // Allocate an array of strings to store the citations.
        m_citations = new string[count];
        m_numCitations = count;
        // Clear the stream state.
        inputFile.clear();
        // Seek back to the start of the citations.
        inputFile.seekg(citationsStart);
        // Read each citation and store it in the new array.
        for (count = 0; count < m_numCitations; ++count) {
            string temp;
            getline(inputFile, temp);
            if (!temp.empty()) {
                m_citations[count] = temp;
            }
        }
    } else {
        m_numCitations = -1;
    }
}
```

Running the same test again on paper1.txt now shows you the correct four citations.

20) Add that “inputFile.clear();” and confirm correct output with all citations.

21) Reflect: Now that we fixed the bug, do you think the tracing is still helpful or it would be better to turn it off?

==================== THE END ======================