Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

21.6 Question 6 (10 marks)

In this exercises you will write C++ code to read in and process Error Code Analysis from a Log File. The details of the functions you will implement are given below. You are provided with the following files:

- `main.cpp` - provided for you to use while you develop your code. This will not be used by the unit tests to test your code
- `task.h` - read only header file
- `task.cpp` - you will write your assessed C++ code in this file
- `utilities.cpp` - (hidden) contains function `print2DVec` which you can use to print your 2D vector - see usage in `main.cpp`

Task 1

Implement the following function:

```
std::vector<std::string> readLogFile(const std::string&
filename);
```

The function has the following parameter:

- `filename` - the names of a log file to read from (format of file shown below)

The function:

- Opens the file to read
  - if the file is not found then the function returns an empty vector (with size 0)
  - if the file is found, then it is opened and each line is read in and stored in a `std::vector` of `string`
- Returns the `std::vector` of `string`

File Format The format of the log file is as follows:

```
2025-03-26T12:00:01 ERROR_404_File_not_found
2025-03-26T12:01:15 WARNING_High_memory_usage
2025-03-26T12:02:30 ERROR_500_Internal_server_error
```

with the following substrings in the string (separated by whitespace):

- a date/time stamp, using the IS0 8601 standard format for dates and times (a concise and unambiguous way to represent date and time in a machine-readable format and where T is used to separate the date and time) e.g. `2025-03-26T12:00:01`
- error/warning messages e.g. `ERROR_404_File_not_found`

NOTE your code will only be tested with log files with lines containing 2 substrings as shown above

---

Task 2

Implement the following function:

```
std::vector<std::vector<std::string>> processLogEntries(const
std::vector<std::string>& logs);
```

This function has the following parameter:

- `logs` - a vector of string, as generated by `readLogFiles` (see above)

The function details are:

- for each string in `logs` it should extract the following as separate strings:
  - date/time stamp
  - error/warning message
- store them in a 2D vector where
  - each row in the 2D vector corresponds to a log entry
  - the columns contain 2 elements, the strings for the date/time stamp and the error/warning message.
- the function returns the 2D vector

NOTE:

- there will only ever be 2 elements (as shown in the examples)
- separated by whitespace
- **HINT:** use sstream to extract from the strings

671842.4329690.qx3zqy7

LAB ACTIVITY | 21.6.1: Question 6 (10 marks) | 0 / 10

Current file: main.cpp | Load default template...

```
1  #include "task.h"
2  #include "utilities.cpp"
3  #include <iostream>
4  #include <vector>
5  #include <string>
6
7  int main() {
8
9      std::string filename = "log.txt";
10     std::vector<std::string> logs = readLogFile(filename);
```

```
10    std::vector<std::string> logs = readLogFile(filename);
11
12    std::vector<std::vector<std::string>> parsedLogs = processLogEntries(log
13    std::cout << print2DVec(parsedLogs);
14
15
16    return EXIT_SUCCESS;
17 }
18
```

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.cpp** (Your program) →

Program output displayed here

Coding trail of your work What is this?

History of your effort will appear here once you begin working on this zyLab.

Trouble with lab?