**CS 124 Spring 2021** **Lab 4 (Optional Pairwork)**

Lab report Part 1 (Purpose, Plan) due Friday April 9, 11:59pm (20 points)
Live Demo Thursday Apr 15, 9am (Both partners must be present. 20 points)
Lab report Part 2 (remaining report sections),source, data files due Friday Apr 16, 11:59pm (60 points)

**Purpose**

This lab assignment is to provide you with an opportunity to:

- demonstrate your ability to write a C++ program utilizing basic C++ concepts (specific concepts should be listed as part of your report)
- demonstrate understanding of concepts covered/reviewed so far in the course (list out specific concepts)

**Lab Requirements**

This lab assignment requires submission of source file(s) and a lab report. For full credit for this lab, these following files must be submitted via Canvas by the assigned due dates:

- Part 1 of the lab report (pdf). This includes the Purpose and Plan sections.
- Part 2 of the lab report (pdf). This includes the remaining sections.
- Live demo as scheduled. Text test files will be provided.
- Source and data file(s) (cpp, txt, zip). Use a part of your name for your files.

**Description**

Write a program that will

- Greet the user, describe your program, and invite the user to try it (user can say no)
- Encode (from English to Morse code) or decode (from Morse code to English) text provided in a data text file. The data provided for input will be in alphabetic English or will be in Morse Code.
    - The morse.txt file provides the cipher: an alphabetic and Morse code equivalence. For any characters not on this cipher list, add them to the list in a rational way. Explain how you decided to add them to the list.
    - Build your own associative array (no STL hash) based on the cipher. Would this be best to use for encoding or decoding? Discuss this, and the key decisions (size of array, hash function, collision management, etc.) in the Development section.
    - Build your own decision tree (no STL) based on the cipher. Would this be best to use for encoding or decoding? Discuss this in the Development section.
    - Input to be translated will be provided in a .txt file.
        - If the input is in alphabetic form, it should be translated into Morse code, with the Morse code equivalent appended to the text file.
        - If the input in in Morse code, it should be translated to the alphabetic equivalent and appended to the text file.

- Create your own text file for testing for inclusion in your report. Discuss the strengths of your test file. (What makes it a good test for your program's capabilities?)

In your program

- Account for appropriate spacing for words and sentences.
- Use iterative loops effectively instead of hard coding the processes of building an associative array or a decision tree
- Use dynamic memory allocation and then deallocate appropriately. Use valgrind to profile your memory usage to ensure no memory leaks at the end of your program. Include this discussion in your Product section.

Take screenshots of your program during development and testing. These may be handy for your report. Use MLA or APA format for citations.

If input text file contains:

```
Life is like a box of chocolates.  You never know what you're
going to get.
```

After your program, the text file will contain:

```
Life is like a box of chocolates.  You never know what you're
going to get.
```

```
.-.. .. ..-. .    .. ...    .-.. .. -.- .    .-    -... --- -..-
--- ..-.    -.-. .... --- -.-. --- .-.. .- - . ... .-.-.-    -.-
- --- ..-    -. . ...- . .-.    -.- -. --- .--    .-- .... .- -
-.-- --- ..- .----. .-. .    --. --- .. -. --.    - ---    --. .
- .-.-.-
```

Or, if the file originally contained Morse code:

```
.-.. .. ..-. .    .. ...    .-.. .. -.- .    .-    -... --- -..-
--- ..-.    -.-. .... --- -.-. --- .-.. .- - . ... .-.-.-    -.-
- --- ..-    -. . ...- . .-.    -.- -. --- .--    .-- .... .- -
-.-- --- ..- .----. .-. .    --. --- .. -. --.    - ---    --. .
- .-.-.-
```

After your program, the text file will contain:

```
.-.. .. ..-. .    .. ...    .-.. .. -.- .    .-    -... --- -..-
--- ..-.    -.-. .... --- -.-. --- .-.. .- - . ... .-.-.-    -.-
- --- ..-    -. . ...- . .-.    -.- -. --- .--    .-- .... .- -
-.-- --- ..- .----. .-. .    --. --- .. -. --.    - ---    --. .
- .-.-.-
```

```
Life is like a box of chocolates.  You never know what you're
going to get.
```

As previous, to install valgrind on Ubuntu, the following commands worked for me. You may need to research for your own environment.

```
sudo apt-get update
sudo apt-get install valgrind
```

To run valgrind on Ubuntu, compile your program and create an executable file. Then run valgrind like below , if your executable is titled `example`

```
valgrind ./example
```

**Lab Report requirements**

The lab report should have these required sections:

- Purpose
- Plan
- Development process
- Product
- Pitfalls
- Possible improvements

The lab report should be submitted as a .pdf in Canvas. You may create the report in a document creation software of your choice. Please ensure that your lab report is cleanly formatted and free of distracting grammatical or other errors. These errors will cost you points on clarity. For more specifics, check Canvas for the grading rubric.

**Purpose**

Provide a few sentences describing the purpose of this assignment, not just the purpose of the program. Include a description of what your program does and the reasons why we are doing this assignment. Please use your own words. (I don't need to read my own words again.) This should be in paragraph form. (A paragraph includes complete sentences, not a collection of phrases that reads like a series of text messages.)

**Plan**

Describe the process you will follow in order to plan out and structure this project for successful completion. The steps here should be specific enough for others to replicate your plan. Provide screenshots, other visuals, urls, etc. as needed. You do not need to reproduce the assignment directions, but provide other information to allow another to be successful in

this assignment. All visuals should be titled and discussed, i.e. not just stuck on the page with no context.

This should be mainly in paragraph form, with lists of steps and visuals as needed. Use your own words. The description does not need to include every detail, but, as mentioned above, should provide enough information so that your planning process can be replicated.

Include a timeline to show your planned activities, what has been accomplished, and the expected timepoints for other activities to lead to the successful completion of your assignment.

At least one flowchart to show program flow is required. The flowchart must have a start and an end. All decision points should be represented as a diamond.

If working with a partner, your timeline should include collaboration times and each partner's activities. Expected contributions of each partner should be clear throughout all the planning.

**Development process**

Describe the process you followed in order to develop the code for this project. Which parts did you choose to develop first? Why? How did you use the plans you developed? How did you test your program to make sure it works correctly?

This should be mainly in paragraph form, with lists of steps and visuals as needed. Use your own words. The description does not need to include every detail, but, as mentioned above, should provide enough information so that your planning process can be replicated.

Individual contributions of each partner should be clear throughout all parts of the development process.

**Product**

Describe your program run(s). Include screenshots for clarity. Discussions should include how the test cases show the capabilities and limitations of your program. See the rubric for more details.

**Pitfalls**

Describe any difficulties and issues you encountered during this assignment. Provide screenshots and other visuals as needed to describe these clearly. Describe how you resolved these issues. If somehow, you didn't have any difficulties, issues, programming errors, or bugs, include an explanation of why this could be the case. See the rubric for more details.

**Possible improvements**

What could you have done differently to improve your completion of this assignment? Be specific. What could be improved about this assignment overall for you and/or for future students? Be as specific and constructive as possible. See the rubric for more details.

**Program file submission requirements**

Submit the files via Canvas

Programming practices expected for this course:

- Good comments to identify the programmer, date, and the project
- Good comments for any significant blocks/lines of the program, comments which explain the purpose of the code and not just what the code does
- Specific functions (at least 5 functions outside of main)
- Do not use “using namespace std”
- Professional and user-friendly interactions and output
- Use local variables – global variables should be used only for constant values
- Be sure your program executes in C++ 11 with no extensions