

---

# ASSIGNMENT 1

---

CS 361: Systems Programming / Spring 2023

## Description

In this assignment, you will explore an ELF object file by listing its functions and global variables, and then calling those functions and modifying those global variables.

## Object file `lib.o`

This object file contains a function responsible for compute the fingerprint of input strings. Unfortunately, the source code for this object file was lost in a recent cyber-attack, and the only thing left is the object file itself. The specification of the fingerprint function is also lost, so we cannot simply re-implement it.

Your job is to use the knowledge gained in CS361 about object files to list all functions and global variables contained in file `lib.o`. One of those functions does not take any argument and does not return anything, it simply prints information about all the other functions and global variables. After understanding which functions and variables are defined inside file `lib.o`, your task should be to figure out which function is this, and write a C program that links against file `lib.o` and calls the function that describes its usage.

The fingerprinting function in file `lib.o` is not perfect, and sometimes fails to compute a meaningful value. Fortunately, other functions and variables in file `lib.o` allow you to detect when the fingerprint fails.

## Format of the input and output

Your program should take the following input as a single argument:

- `functions`
  - List all the functions in file `lib.o` in alphabetical order
- `variables`
  - List all the variables in file `lib.o` in alphabetical order
- `help`
  - Call the function inside `lib.o` that prints a helpful message about how all functions and variables are supposed to be used
- `<anything else>`
  - Compute the fingerprint of the argument provided
  - If file `lib.o` fails to compute the fingerprint, simply print the argument back

## Entry Point

You will modify files `main.c` and `Makefile`. File `main.c` will contain your implementation, file `Makefile` must be modified to link your solution against file `lib.o`.

## Due Date and Resubmission Policy

This assignment is due on **January 28<sup>th</sup> 2023** (Saturday) at **5pm CST**. There is no late policy.

The code and date used for your submission is defined by the last commit to your Git repository.

To resubmit this assignment, you are required to have attendance to Lab Session 1. You can resubmit your assignment until **February 4<sup>th</sup> 2023** (following Saturday) at **5pm CST**. Together with your resubmission, you will have to submit a written description of what you changed from the original submission (on Gradescope).

## Bonus Points

This assignment has a total of **15% bonus points**:

- 10% can be earned by using Piazza as described in the syllabus. Your posts should be public, tagged with the `a1` label, and non-anonymous to the instructors to count towards the bonus. You can claim bonus points through [a Gradescope quiz](#).
- 5% can be earned by completing Lab Session 1.

## Submission and Grading

This assignment is submitted through Github, and has an automatic grade component of 100%. You can check your current grade at any point by submitting your code and checking the autograder. The automatic grade is determined by 10 tests, that will check if your submission outputs the expected result. Each test is worth 10%.

## Errors and Omissions

If you find an error or an omission, please post it on Piazza as soon as you find it.

## Hardcoding and Academic Integrity

Any hardcoding will result in a 0% grade. Hardcoding is when you submit code that detects which test is being run, and simply outputs the expected result. For instance, detecting that test 22 is running, and replacing the usual execution of your submission with `printf("expected result")`.

The academic integrity policy described in the syllabus applies to this assignment. You are responsible for writing all the code that you submit. We will use an automatic tool that detects plagiarism on all submitted code, and we will investigate all instances where plagiarism is more than likely.

Please refer to the syllabus for the full academic integrity policy.