

# ICS 46 (Shindler) Fall 2022

## Project #0: Getting to Know the ICS 46 VM

**Due date and time: Friday September 30, 2022 at 7:30 AM.**

Remember to read the syllabus to be sure you make yourself eligible to be graded for this assignment.

If you took Professor Thornton's ICS 45C, the VM should look *very familiar* to you. If you still have VMWare, you may use that for the VM. If you do not, however, we would like you to use VirtualBox to host the VM. Be advised that this is not the same thing as Project #0 from when you were in that class, so you still have some work to do.

In fact, many of this assignment instructions are drawn from his similar assignment for ICS 46<sup>1</sup> and he was instrumental in helping me set up the ICS 46 VM. That having been said, you should still read these directions, as they may have changed from when you took ICS 45C, even if he was your teacher for that class.

Remember:

- You need to download the VM for this class, not the one for Prof. Thornton's ICS 46
  - <https://www.ics.uci.edu/~mikes/ics46/vm/ics46-f22.ova> (x86)
  - <https://www.ics.uci.edu/~mikes/ics46/vm/ics46-f22-arm-vm.tar.gz> (M1)
- You need to do this project, not the one for one of Professor Thornton's classes. Just because the VMs look similar doesn't mean we're giving the same assignments.

For students who wish to work in a quiet environment on campus, I recommend loading the VM onto a USB "thumb" drive and taking it to the ICS third floor computer lab. I also recommend you be very careful in backing up your work in progress, no matter where you are doing this, but especially if you are working via a thumb disk in a computer lab.

### The Program's Requirements

As a warm-up, this project asks you to write and submit a short C++ program. The program itself isn't actually the interesting part, though it's one that you might find takes you a little bit of time to write. The main goal here is to be sure you're able to use the ICS 46 VM to do your work, that you learn what you need to know about one of the available text editors to write your program, and that you use the provided tools to gather your files for submission. Even if you normally prefer a different working environment, you would be well-served to use the ICS 46 VM for this project, to be sure that you can use it for your work later in the quarter. Also, if your project does not compile and run in the VM, we will not be able to grade it, and that will cause you to get a zero. Every quarter this happens to some students who, to put it gently, are not pleased with their grade on any such project. Please do not let this happen to you!

See the lab manual for information on how to set up the VM and how to start a project for this class.

For more information about grading, see the relevant section of the ICS 46 lab manual.

---

<sup>1</sup> You know how I said that if you get significant but allowable help from someone else on an assignment, you should cite who and what clearly? That rule applies to me too.

## The Program's Requirements

Consider combinatorial puzzles, like those presented in lecture during lecture two. For example, one such puzzle is POT + PAN = BIB. In these puzzles, you must assign each letter a distinct digit, such that if we substitute the digits in place of the letters, the resulting mathematical equation is true.

In our continuing example, if we set P=2, O=3, T=1, A=7, N=4, B=5, and I=0, we get  $231 + 274 = 505$ . In the file proj0.cpp, you will find an incomplete implementation for:

```
bool verifySolution(std::string s1, std::string s2, std::string s3,
const std::unordered_map<char, unsigned> & mapping)
```

The first part of your assignment is to finish implementing this. To find out what the proposed mapping has for  $s1[0]$  (the first character in std::string  $s1$ ), you can use the member function `at()` of the `unordered_map` class: that is, `mapping.at(s1[0])`.

You **are not** required to check that the digits assigned are distinct, merely that it satisfies the equation. Note that  $s1$ ,  $s2$ , and  $s3$  might not be equal in size. Furthermore, you may assume that all characters appearing in any of strings  $s1$ ,  $s2$ , and/or  $s3$  will appear in the mapping; you do not need to check for that.

There are many ways to solve this problem. If you are having trouble finding *any* way to write a solution to it, please see a member of course staff. You may not use the standard library for any portion that solves a significant portion of your assignment, other than making use of accessor methods in the `std::unordered_map` class. Standard library portions that do not solve a significant part of the assignment, such as the `std::atoi()` function, are allowed. You are not required to use that function and many correct solutions without it exist.

If you want to test, and you should, two test cases have been provided in the gtest folder. Build your project (either fully or just the testing portion) and run the testing portion to see. While I will have other tests when I grade it, I promise these two will be part of it. *In general, I will provide you with some test cases for every project. However, it is not guaranteed that passing these cases is sufficient for a 100% on the assignment. You still need to develop your own test cases. Testing your code is an important part of the assignment, not a mere afterthought.*

When we grade this, any test case that takes more than two minutes to run on the instructor's (reasonably modern) computer may be deemed an unsuccessful test case.

## **Deliverables**

See the ICS 46 Lab Manual for information about submitting projects. We will grade only what was submitted before the deadline. If you replaced some of your files with newer versions before the deadline, we will grade only the most recent submission of each unless you fill out the relevant form *before* the submission deadline.

We will not grade files submitted after the deadline has passed, nor will we grade files submitted via email or in paper form.

You are responsible for submitting the version of your project that you want graded. We will grade only what you submitted before the deadline. Accidentally submitting the wrong version, or forgetting to submit files, will not be considered grounds for a regrade.

This project is not included in the late work policy for this course. **It needs to be completed on schedule and submitted before the due date above.** Submissions beyond that deadline will not be considered.

## **Your grade on this project**

There are 2 points possible on this project; they are only available by test cases. We will run test cases with the code you submit; each test case is worth some fraction of the grade. Test cases that take longer than two minutes to run on the instructor's (reasonably modern) computer may be deemed incorrect runs, even if a longer amount of time available to them would cause a correct answer.

For more information about grading, see the relevant section of the ICS 46 lab manual.