CS50 Section Notes

Zad Chin

Table of contents

Preface

This website contains section notes for COMPSCI 50, an Introduction to Computer Science course at Harvard University taught by Professor David Malan. These notes are created by Zad Chin.

Logistics

- Course Website
- Sections: 7:00 - 8:20 pm EDT on Fridays at Zoom
- Office Hours: 7:00 - 8:20 pm EDTon Fridays at Zoom
- Manual Pages : website here
- EdStem Discussion: Ed link

Notes

These section notes will be presented as an online book, and the source for this book at https://zadchin.github.io/CS50Section/. Any typos or errors can be reported at https://github.com/zadchin/CS50Section/issues. Thanks for reading.

This is a Quarto book. To learn more about Quarto books visit https://quarto.org/docs/books.

Section 1

Last Updated: 27 OCT 2023

Date: 27 Oct 2023

This Week

In this section, we will discuss:

- Variables
- Types
- Loops
- Conditions
- Functions

Slides

Slides deck are available here: Slide Section 1

Next Week

Next week, we will discuss:

- Array
- String
- Command Line Argument

Section 2

Last Updated: 10 Nov 2023

Date: 10 Nov 2023

This Week

In this section, we will discuss:

- Array
- String
- Command Line Argument

Slides

Slides deck are available here: Slide Section 2

Section Practice Problems

Section Problem 1

Background

In a classroom, a teacher keeps track of the attendance of students. Each student is assigned a seat number, and their attendance status is recorded as present (1) or absent (0) for a particular day.

Task

Write a program in C that allows the teacher to enter the attendance status for each student and then displays the total number of students present and absent on that day.

Demo

Demo in Section

Starter Code

Copy and following the following code to a new C file in CS50 codespace to start coding!

```
#include <cs50.h>
#include <stdio.h>

#define CLASS_SIZE 5

int main(void) {
    int attendance[CLASS_SIZE];
    int present = 0;
    int absent = 0;

    // Ask the teacher to enter attendance
    printf("Enter the attendance for each student (1 for present, 0 for absent):\n");
    for (int i = 0; i < CLASS_SIZE; i++) {
        // Use get_int() to get input from user
        attendance[i] = get_int("Student %d: ", i + 1);

        // TODO: Add code to count presents and absents
    }

    // TODO: Add code to display the total presents and absents

    return 0;
}
```

Solution

Solution will be released after section!

Section Problem 2

Background

Gen Z often use acronyms to help remember lists or sequences, for example:

- YOLO- You only live once
- FOMO – Fear of missing out
- GOAT – Greatest of all time

Let's create a program that generates an acronym from a list of words.

Task

Write a program in C that:

Asks the user to input a certain number of words, stored in an array of strings. Generates an acronym by taking the first letter of each word and concatenating them. Converts the acronym to uppercase. Prints out the final acronym.

Demo

Demo in Section

Starter Code

Copy and following the following code to a new C file in CS50 codespace to start coding!

```
#include <cs50.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define WORD_COUNT 4 // Number of words in the acronym

int main(void) {
    string words[WORD_COUNT];

    //TODO: Initialize acronym

    // Prompt user for words
    for (int i = 0; i < WORD_COUNT; i++) {
        words[i] = get_string("Enter word %d: ", i + 1);
    }

    // TODO: Create and print the acronym

    return 0;
}
```

Solution

Solution will be released after section!

Section Problem 3

Background and Task

For the acronym generator, the words that form the acronym can be passed as command-line arguments. Recode the origin code files to take words from command-line arguments to create the acronym.

Demo

Demo in Section

Starter Code

Copy and following the following code to a new C file in CS50 codespace to start coding!

```
#include <cs50.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main(int argc, string argv[]) {
    // TODO: Fill up ... to make sure users give at least a certain argc
    if (...) {
        printf("Usage: ./acronym word1 word2 ... wordN\n");
        return 1;
    }

    // TODO: Create and print the acronym from command-line arguments

    return 0;
}
```

Solution

Solution will be released after section!

Next Week

Next week, we will discuss:

- Structures in C

- Sorting
- Searching
- Recursion

Note: I am out for the next section.Margaret Tanzosh will replace me for the section. Refer her materials.