

This is CS50.

Lab 2

Hey hey!

Name: Zad

Class: 2024 (Junior) @ Adams

Concentration: CS + Math

Passionate about: Teaching &
Research & Startup!

Free time: Work on Startup, Watch
Documentary / Anime



**Any questions for
me?**

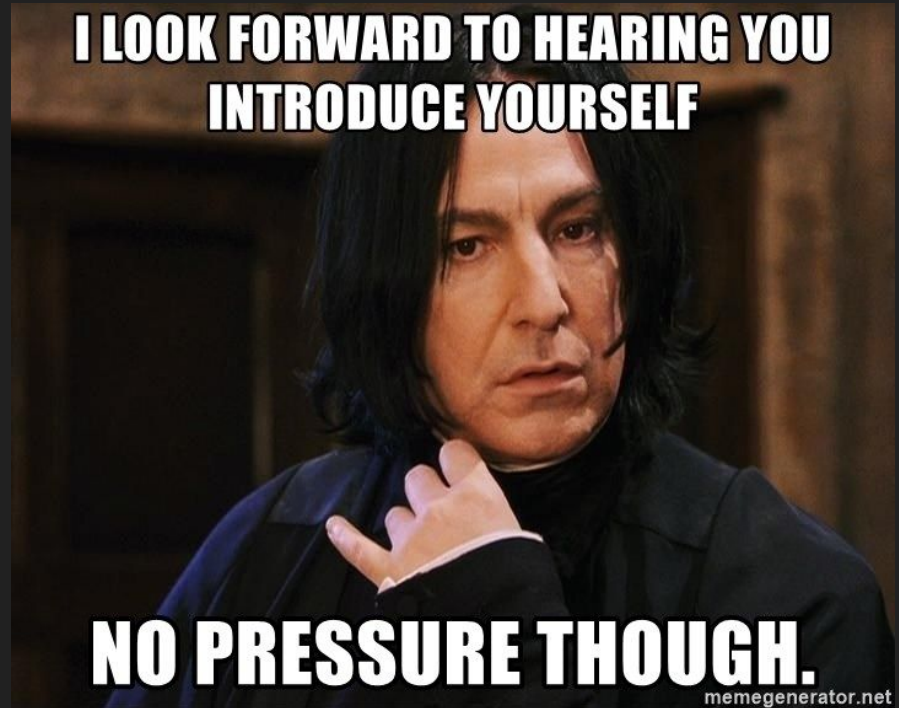
Let me get to know you too 🙄

Name

Pronoun

Concentration

Toilet Roll time! :P



You need to know that:

IT IS OKAY to feel INSECURE.

IT IS OKAY to feel INFERIOR.

IT IS OKAY to feel VULNERABLE.

Reach out!



- Me @ +18579288166/
zadchin@college
- Heads of CS50 @
heads@cs50
- Ed, Office Hours, Tutorial
- **Your friends and peers
in this group and class**

Customizing our lab:

- Review confusing concepts
- Review previous related exam questions to prepare you for the test
- Talk about psets, where we will walkthrough it together
- Laugh, talk, have fun together!
- ... and much more



I want to hear from you...



**Any questions for
me?**

Pset 1 Quick Review

**It is okay not to get 5/5
from design score! 4/5 is
amazing!**

Design thinking:

- Could you eliminate:
 - Repetition?
 - Loops?
 - Magic Numbers?
- Is the code consistent?
- Is the code organized logically and well-documented?
- Is there a shorter way to accomplish the same thing?
- Is there a more efficient way to accomplish the same thing?
- Should the code be in its own function?
- Are variables given descriptive names?

Example: Calculating Quarters (Cash)

```
int calculate_quarters(int cents)
{
    int quarters = 0;
    while (cents ≥ 25)
    {
        cents = cents - 25;
        quarters++;
    }
    return quarters;
}
```

```
int calculate_quarters(int cents)
{
    int quarters = cents / 25;
    return quarters;
}
```

Is there a one line solution?

Yes!

```
int calculate_quarters(int cents)
{
    return cents / 25;
}
```



Ele-le... Ele-le-le-le...

**Any questions for
me?**

Concept Deep Dive

Week 2 Concepts:

- **Array**
- **String**
- **Command-Line Argument**

ARRAYS

What is an array?

A data type that allows us to store multiple values of the same type in memory

ARRAYS

What is an array?

A data type that allows us to store multiple values of the same type in memory

Why must arrays contain values of the same type?

ARRAYS

What is an array?

A data type that allows us to store multiple values of the same type in memory

Why must arrays contain values of the same type?

C has only partitioned enough memory for that particular type and size of array!

ARRAYS - HOW DO WE DECLARE THEM?

```
type name[size];
```

```
Ex: int nums [5];
```

ARRAYS - HOW DO WE INITIALIZE THEM?

- You can either initialize with declaration (instantiation):

```
type name[size] = {<values>;
```

Declaration

Initialization


Ex: `int nums[3] = {50, 55, 120}`

ARRAYS - HOW DO WE INITIALIZE THEM?

- Or you can initialize separately:

```
Ex: int nums[3];  
    nums[0] = 50;  
    nums[1] = 55;  
    nums[2] = 120;
```

Note that it starts with index 0! **Arrays are always zero-indexed!**



**Any questions for
me?**

SO... WHAT IS A STRING?

Strings as we've dealt with so far were implemented by the CS50 library. But in reality, they are actually an array of characters.

```
string s = "Hello";
```

H	e	l	l	o	\0
0	1	2	3	4	5

INDEXING INTO ARRAYS

```
string s = "Hello";
```

H	e	l	l	o	\0
0	1	2	3	4	5

s[0]

Remember that arrays are indexed at 0!

COMBINING STRINGS AND ARRAYS!

Consider the following code:

```
string words[2];
```

```
words[0] = "HI!";
```

```
words[1] = "BYE!";
```

What do I have if I print `words[0][0]`? Or

`words[1][4]`?

COMBINING STRINGS AND ARRAYS!

Representations:

H	I	!	\0	B	Y	E	!	\0
words[0][0]	words[0][1]	words[0][2]	words[0][3]	words[1][0]	words[1][1]	words[1][2]	words[1][3]	words[1][4]

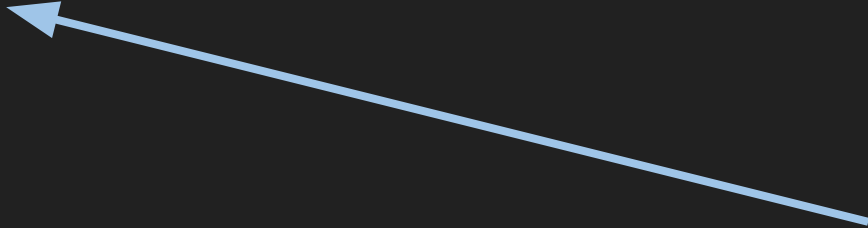
**Any questions for
me?**

REVISITING THE MAIN FUNCTION

```
int main(void)
{
}

```

Does the input
always have to be
void?



REVISITING THE MAIN FUNCTION

```
int main(void)
{
}

```


Does the input
always have to be
void?

NO!

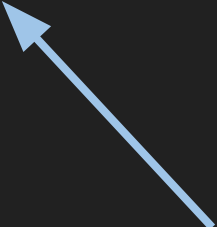
COMMAND LINE ARGUMENTS

We can use command line arguments to pass arguments into our program:

```
int main(int argc, string argv[]) {  
  
}
```



**argc represents the
number of arguments
we've passed via the
command line**



**argv is an array of
strings with the
different command line
arguments**

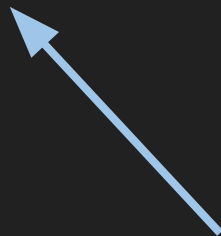
COMMAND LINE ARGUMENTS

```
$ ./main "x" "y" "z"
```

COMMAND LINE ARGUMENTS

```
$ ./main "x" "y" "z"
```

argc would be 4



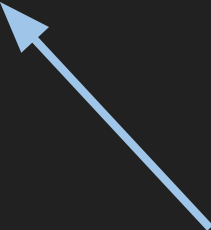
argv[0]	argv[1]	argv[2]	argv[3]
"main"	"x"	"y"	"z"

COMMAND LINE ARGUMENTS

```
$ ./main "x" "y" "z"
```

argc would be 4

argv[0] is always the
name of the program



argv[0]	argv[1]	argv[2]	argv[3]
"main"	"x"	"y"	"z"

**Any questions for
me?**

Lab Time!

Task:

- 1) Print winner based on score computed
- 2) Compute score based on the POINTS[] defined

2) Compute score based on the POINTS[] defined

Imagine you have the following word “Hello”:

H	e	l	l	o	\0
0	1	2	3	4	5

How would you count the score of this word?

2) Tips:

- Use for loop over length of the word
- Beware of uppercase and lowercase! Use `isupper` and `islower` to check!
- Try to explore how to get the index of an alphabet by + / - from 'A' or 'a'!

**Any questions for
me?**

Let's do lab!

Discussion (1): Print winner based on score computed

```
// Print the winner
if (score1 > score2)
{
    printf("Player 1 wins!\n");
}
else if (score1 < score2)
{
    printf("Player 2 wins!\n");
}
else
{
    printf("Tie!\n");
}
```

Discussion (2): Compute score based on the POINTS[] defined

```
int score = 0;

// Compute score for each character
for (int i = 0, len = strlen(word); i < len; i++)
{
    if (isupper(word[i]))
    {
        score += POINTS[word[i] - 'A'];
    }
    else if (islower(word[i]))
    {
        score += POINTS[word[i] - 'a'];
    }
}

return score;
```

Previous related Test Questions

Fair and Balanced (2017 Test)

<https://tinyurl.com/CS50-Lab2-Q1>

Let's consider an array "balanced" if the sum of the elements on the **left half of the array is equal to the sum of the elements on the right half of the array**. In an array with an odd number of elements, we ignore the element in the middle when making that determination.

Fair and Balanced (2017 Test)

Example: `int a[] = {17, 40, 28, 29};`

17	40	28	29
----	----	----	----

is **balanced**, because the sum of its left half is $17 + 40 = 57$, just as the sum of its right half is $28 + 29 = 57$.

Fair and Balanced (2017 Test)

Example: `int b[] = {30, 22, 11, -14, 66};`

30	22	11	-14	66
----	----	----	-----	----

is **balanced**, because the sum of its left half is $30 + 22 = 52$, just as the sum of its right half is $-14 + 66 = 52$; we ignore its middle element (11) altogether.

Fair and Balanced (2017 Test)

Q1: Is the array below balanced?

16	26	39	3	39
----	----	----	---	----

Yes!

Fair and Balanced (2017 Test)

Q2: Is the array below balanced?

0	0	0	0	0
---	---	---	---	---

Yes!

**Any questions for
me?**

Fair and Balanced (2017 Test)

Q3: Implement a function called `balanced` that accepts two parameters:

- **an array of integers, `array`,**
- **an integer, `n`, that represents the length of array.**

The function should return `true` if `array` is balanced and `false` otherwise. Assume that `n` is positive.

Fair and Balanced (2017 Test)

Starter code:

```
int main(void)
{
    int array[5] = {16, 26, 39, 3, 39};
    int n = 5;
    /* code goes here */
}
```

Fair and Balanced (2017 Test)

Answer:

<Refer Github!>

**Any questions for
me?**

Problem Set Tips

Problem Set: Readability

Prompt walkthrough, watch Bryan's Video




Useful tips (1):

- In `count_words`, you need to need to check whether the words are “alphabets”, ie, you don’t want to count 3 or 4 as words.
- You can use `ASCII`, but it will be chunky;
- Hence, consider using `isalpha` from CS50 manual;


How to use `isalpha`?

```
# include <ctype.h>
int main(void)
{
```

Remember to include
the library that contains
`isalpha`!



```
    char c = get_char("Input: ");
    if (isalpha(c))
    {
        printf("Your input is alphabetical.\n");
    }
    else
    {
        printf("Your input is not alphabetical.\n");
    }
}
```



Notice that `isalpha`
takes in characters!
You will need to index
into the string!

- More documentation at <https://manual.cs50.io/3/isalpha>

```
# include <ctype.h>
// count words
Float count_words(string words){
    // index into every character in words and count character if
    isalpha return True
    for i=0; i <strlen(word); i++){
        If (isalpha(word[i])){
            count+=1
        }
    }
}
```

Useful tips (2 & 3):

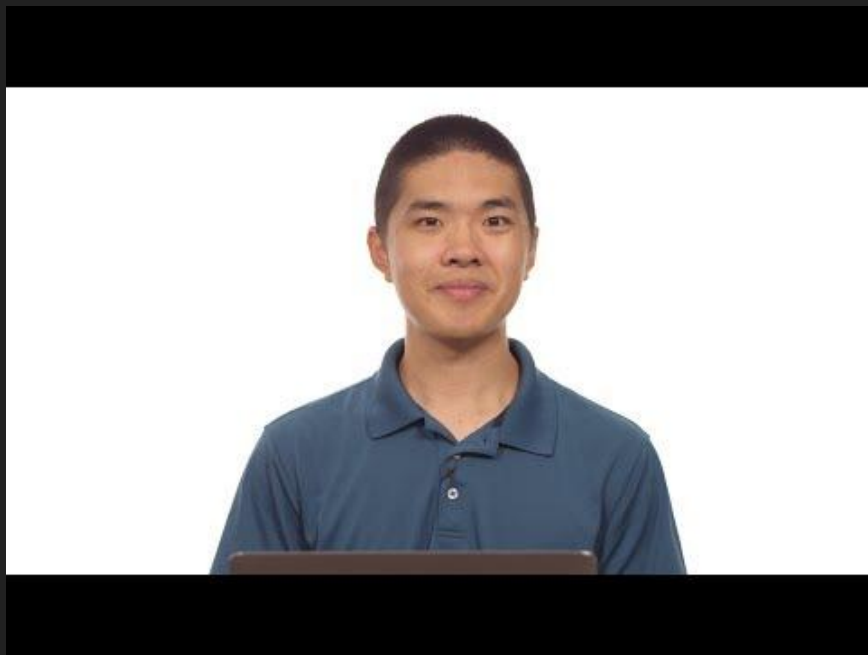
- Please consider what are the ways to count words that does not assume one space per word and does not initialize counting at one!
- Break into lines for long code (calculating Coleman index)

```
int a = a + b + c + d / e + f...
```

```
int a = a + b + c + d / e + f  
      + g..
```

Problem Set: Caesar

Prompt walkthrough, watch Bryan's Video



Pseudocode

```
// initialize magic number ALPHABET_SIZE = 26 (to be use later!)  
  
// check if sufficient argument (argc == 2)  
  
// check if every char in argv[1] is a digit  
  
// change argv[1] from string to digit <atoi> (this is the key)  
  
// check whether key is negative  
  
// ask user for string with get_string  
  
// change each character in with magic formula that involve above  
ALPHABET_SIZE and ASCII value of 'A' and 'a'! Remember to check  
it is uppcase or lowercase before that!
```



Useful tips (1):

- In validating keys, you need to check whether the argument is a digit
- You can use ASCII, but it is not that clean
- Hence, consider using `isdigit` from CS50 manual;


How to use `isdigit`?

```
# include <ctype.h>
int main(void)
{
    char c = get_char("Input: ");
    if (isdigit(c))
    {
        printf("Your input is a digit.\n");
    }
    else
    {
        printf("Your input is not a digit.\n");
    }
}
```

Remember to include
the library that contains
`isdigit`!



Notice that `isdigit`
takes in characters!
You will need to index
into the strings!



- More documentation at <https://manual.cs50.io/3/isdigit>

Useful tips (2 & 3):

- In checking whether a char is uppercase or lowercase, consider using `isupper` and `islower` from the library `<ctype.h>`
- Instead of comparing the character to ASCII number, you can directly compare it:

```
if c < 65:  
    printf("Yayy")
```

```
ans = c - 65
```

```
if c < 'A':  
    printf("Yayy")
```

```
ans = c - 'A'
```

Thank you!

Feedback form:



tinyurl.com/zad-feedback

See you next week!