This is CS50

CS50's Introduction to Computer Science

OpenCourseWare

Donate (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.clubhouse.com/@davidjmalan) (https://www.facebook.com/dmalan) (https://github.com/dmalan) (https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/) (https://orcid.org/0000-0001-5338-2522) (https://www.quora.com/profile/David-J-Malan) (https://www.reddit.com/user/davidjmalan) (https://www.tiktok.com/@davidjmalan) (https://davidjmalan.t.me/) (https://twitter.com/davidjmalan)

Snackbar

Learning Goals

- Practice using `struct`s
- Write a linear search algorithm

Background

Imagine you're at the beach and want to order a number of items from the snack bar. You have a limited amount of cash on you, and you want to get a total cost for your items before ordering. In `snackbar.c` you will complete two functions. First is `add_items` which will add at least the first 4 items on the Beach Burger Shack menu. Then you will complete `get_cost` which will implement a linear search algorithm to search for each item you choose, and return the corresponding price.

⊞ Hints

Demo

Getting Started

1. Log into code.cs50.io (https://code.cs50.io/) using your GitHub account.
2. Click inside the terminal window and execute `cd`.
3. Execute `wget https://cdn.cs50.net/2022/fall/labs/3/snackbar.zip` followed by Enter in order to download a zip called `snackbar.zip` in your codespace. Take care not to overlook the space between `wget` and the following URL, or any other character for that matter!
4. Now execute `unzip snackbar.zip` to create a folder called `snackbar`.
5. You no longer need the ZIP file, so you can execute `rm snackbar.zip` and respond with "y" followed by Enter at the prompt.

Implementation Details

The `main` function is already complete. After calling `add_items` to initialize the `menu` array, it will print out the menu items and their prices, prompting you to keep selecting items until you press enter without typing anything in. You are to complete two functions, `add_items`, which adds at least the first four menu items, and `get_cost` to return the cost of each item. When you are creating a linear search algorithm in `get_cost`, do make sure that it is case insentive.

Thought Question

- Why is an array of `struct`s a better solution here than multiple arrays?

How to Test Your Code

Your program should behave as follows:

```
snackbar/ $ ./snackbar

Welcome to Beach Burger Shack!
Choose from the following menu to order. Press enter when done.

Burger: $9.50
Vegan Burger: $11.00
Hot Dog: $5.00
Cheese Dog: $7.00
Fries: $5.00
Cheese Fries: $6.00
Cold Pressed Juice: $7.00
Cold Brew: $3.00
Water: $2.00
Soda: $2.00

Enter a food item: burger
Enter a food item: fries
Enter a food item: soda
Enter a food item:

Your total cost is: $16.50
```

```
snackbar/ $ ./snackbar

Welcome to Beach Burger Shack!
Choose from the following menu to order. Press enter when done.

Burger: $9.50
Vegan Burger: $11.00
Hot Dog: $5.00
Cheese Dog: $7.00
Fries: $5.00
Cheese Fries: $6.00
Cold Pressed Juice: $7.00
Cold Brew: $3.00
Water: $2.00
Soda: $2.00

Enter a food item: cold brew
Enter a food item: hot dog
Enter a food item:

Your total cost is: $8.00
```

Note that the menu should only print out the items that you saved in the `menu` *array.*

No `check50` for this one!

To evaluate that the style of your code, type in the following at the `$` prompt.

```
style50 snackbar.c
```

How to Submit

No need to submit! This is an optional practice problem.