Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

11.8 "Save" menu option

We hope that by now you realize the importance of storing and retrieving data. It will help you resume your work from where you left without hardcoding those values. We will now add options that will let you store your tasks into a file and read them back into your task manager.

You need to complete the function `save_tasks_to_csv()` as defined below. You may refer to your `save_temperature_statistics()` function from LA 10.9. Both of these functions perform similar activity. The major change is in the data written in each line of the csv file.

To implement this function, we need to use `import csv` in our **tasks_functions.py** file. The function then uses the csv writer object to write this list as a line into the `filename` file.

```
def save_tasks_to_csv(tasks_list, filename):
    """
    param: tasks_list - The list of the tasks stored as
dictionaries
    param: filename (str) - A string that ends with '.csv' which
represents
                the name of the file to which to save the tasks.
This file will
                be created if it is not present, otherwise, it
will be overwritten.

    The function ensures that the last 4 characters of the
filename are '.csv'.
    The function requires the `import csv`.

    The function will use the `with` statement to open the file
`filename`.
    After creating a csv writer object, the function uses a `for`
loop
    to loop over every task in the list and creates a new list
    containing only strings - this list is saved into the file by
the csv writer
    object. The order of the elements in the list is:

    * `name` field of the task dictionary
    * `info` field of the task dictionary
```

```
    * `priority` field of the task as a string
    (i.e, "5" or "3", NOT "Lowest" or "Medium")
    * `duedate` field of the task as written as string
    (i.e, "06/06/2022", NOT "June 6, 2022")
    * `done` field of the task dictionary

    returns:
    -1 if the last 4 characters in `filename` are not '.csv'
    None if we are able to successfully write into `filename`
    """
```

The portion of the **main program** code is provided below. Complete the missing parts and add them in the correct place to your task manager.

```
    elif opt == 'S':
        continue_action = ...
        while continue_action == 'y':
            print("::: Enter the filename ending with '.csv'.")
            filename = input("> ")
            ... = save_tasks_to_csv(..., ...) # TODO: Call the
function with appropriate inputs and capture the output
            if ... == -1: # TODO
                print(f"WARNING: |{...}| is an invalid file
name!") # TODO
                print("::: Would you like to try again?", end="
")
                continue_action = input("::: Enter 'y' to try
again.\n> ")
            else:
                print(f"Successfully stored all the tasks to |
{...}|")
    #-----------------------------------------------------------
-------------
```

Below are the sample interactions with the user to store tasks into a csv file. The ellipsis is **NOT** a part of the output.

```
You selected option S to > Save the data to file.
::: Enter the filename ending with '.csv'.
> my-tasks
WARNING: |my-tasks| is an invalid file name!
::: Would you like to try again? Enter 'y' to try again.
> y
::: Enter the filename ending with '.csv'.
> my-tasks.csv
Successfully stored all tasks to |my-tasks.csv|
::: Press Enter to continue
```

```
==========================
What would you like to do?
...
```

In he first attempt, the input does not end with `.csv` which is why the program asked the user to try again. The second input ends with `.csv`, so the program writes our tasks to the provided file and shows the success message.

NOTE: This option and the next option should be compatible with each other: i.e., whatever you write to a csv file using this option, you should be able to read back using the next option and vice-versa.

398092.2571084.qx3zqy7

LAB ACTIVITY | 11.8.1: "Save" menu option | 0 / 1

main.py

```
1
```

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.py** (Your program)

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin
working on this zyLab.
```

Trouble with lab?