markus_afonso_as1

Script one:

Part 1 (writetask)

If an option (date, tag, description) is not provided, the script will prompt the user
options should be wrapped in ' ' if they include spaces
Every task must have a description
All valid dates are converted to MMM-dd-yyyy (March 3, 2023 ⇒ Mar-03-2023)

Tasks are stored as:

```
Tag(s):   Date:  Task:
---
Tag(s):  Date:
Task:
line1
line2
line3
...
---
```

help message:

```
vagrant@ubuntu2210:~/data/as1$ ./writetask -h
Usage: ./writetask [-d <due-by-date>] [-t <tag>] <task>
  -d <due-by-date>  Specify a valid date for the task
  -t <tag>          Specify a tag for the task
  -h                Show this help message
```

Adding task with all options

```
vagrant@ubuntu2210:~/data/as1$ ./writetask -d 'march 6 2023' -t bcit write notes for
Task added successfully.
```

```
# in the task file:
# Tag(s): [bcit]  Date: Mar-06-2023 Task: write notes for 2420
```

Adding task with multiple tags

```
vagrant@ubuntu2210:~/data/as1$ ./writetask -t 'bcit notes' -d 'march 6 2023' write n
Task added successfully.

# in the task file:
# Tag(s): [bcit] [notes]  Date: Mar-06-2023 Task: write notes
```

Adding task with only a date and desciption (tag not provided, so user is prompted)

```
vagrant@ubuntu2210:~/data/as1$ ./writetask -d mar-08-2023 buy apples
Enter tag (optional, separate multiple tags with spaces):
Task added successfully.

# in the task file:
# Tag(s):  Date: Mar-08-2023 Task: buy apples
```

Adding task with only a tag and description (date not provided, so user is prompted)

```
vagrant@ubuntu2210:~/data/as1$ ./writetask -t groceries buy milk
Enter due-by date (optional, must be a valid date):
Task added successfully.

# in the task file:
# Tag(s): [groceries]  Date:  Task: buy milk
```

Adding task only a desciption (date and tag not provided, so user is prompted)

```
vagrant@ubuntu2210:~/data/as1$ ./writetask buy milk at the grocery store
Enter due-by date (optional, must be a valid date):
Enter tag (optional, separate multiple tags with spaces):
Task added successfully.

# in the task file
# Tag(s):  Date:  Task: buy milk at the grocery store
```

Adding multiline task (without date and tag)

```
vagrant@ubuntu2210:~/data/as1$ ./writetask
Enter due-by date (optional, must be a valid date):
Enter tag (optional, separate multiple tags with spaces):
```

```
Enter task description (press Ctrl+D to finish):
groceries
        milk
        apples
        kale
Task added successfully.

# in the task file
# Tag(s):  Date:
# Task:
# groceries
#       milk
#       apples
#       kale
```

Adding multiline task (with date and tag)

```
vagrant@ubuntu2210:~/data/as1$ ./writetask
Enter due-by date (optional, must be a valid date):march 6 2023
Enter tag (optional, separate multiple tags with spaces): groceries
Enter task description (press Ctrl+D to finish):
groceries
        milk
        apples
        kale
Task added successfully.

# in the task file
# Tag(s): [groceries]  Date: Mar-06-2023
# Task:
# groceries
#       milk
#       apples
#       kale
```

If user inputs incorrect date returns err

```
# in prompt
vagrant@ubuntu2210:~/data/as1$ ./writetask -t tag description
Enter due-by date (optional, must be a valid date):date
[2023-03-05T20:24:55+0000]: Invalid date format: date
Try -h for help
# in option
vagrant@ubuntu2210:~/data/as1$ ./writetask -t tag -d 'march 32 2023'
```

```
[2023-03-05T20:27:06+0000]: Invalid date format: march 32 2023
Try -h for help
```

Part 2 (readtask)

Like writetask, all dates are converted to MMM-dd-yyyy.
all outputs are returned in a pager

help message:

```
vagrant@ubuntu2210:~/data/as1$ ./readtask -h
Usage: ./readtask [-d <due-by-date>] [-t <tag>]
  -d <due-by-date>  Display tasks due by the specified date in the format of MMM-dd-
  -t <tag>          Display tasks with the specified tag
  -h                Show this help message
```

Show task file

```
vagrant@ubuntu2210:~/data/as1$ ./readtask
```

Current task file

```
Tag(s): [bcit]  Date: Mar-06-2023 Task: write notes for 2420
---
Tag(s):  Date: Mar-08-2023 Task: buy apples
---
Tag(s): [groceries]  Date:  Task: buy milk
---
Tag(s):  Date:
Task:
groceries
        milk
        apples
        kale
---
Tag(s):  Date:  Task: buy milk at the grocery store
---
Tag(s): [groceries]  Date: Mar-06-2023
```

```
Task:
groceries
        milk
        apples
        kale
---
Tag(s): [groceries]  Date:  Task: buy milk near bcit
---
Tag(s): [bcit] [notes]  Date: Mar-06-2023 Task: write notes
---
(END)
```

Search by tag

```
vagrant@ubuntu2210:~/data/as1$ ./readtask -t 'bcit'
```

returns

```
Tag(s): [bcit]  Date: Mar-06-2023 Task: write notes for 2420
---
Tag(s): [bcit] [notes]  Date: Mar-06-2023 Task: write notes
---
(END)
```

Search by date

```
vagrant@ubuntu2210:~/data/as1$ ./readtask -d 'march 8 2023'
```

returns

```
Tag(s):  Date: Mar-08-2023 Task: buy apples
---
(END)
```

Search by tag and date

```
vagrant@ubuntu2210:~/data/as1$ ./readtask -d 'march 6 2023' -t bcit
```

returns

```
Tag(s): [bcit]  Date: Mar-06-2023 Task: write notes for 2420
---
Tag(s): [bcit] [notes]  Date: Mar-06-2023 Task: write notes
---
(END)
```

Script two:

add alias

```
vagrant@ubuntu2210:~/data/as1$ alias rm='/home/vagrant/data/as1/logrm'
vagrant@ubuntu2210:~/data/as1$ rm
Usage: /home/vagrant/data/as1/logrm [-s] <file1> [<file2> ...]
  -s  Delete file(s) silently without writing to the log
  -h  Show this help message
```

help message:

```
vagrant@ubuntu2210:~/data/as1$ ./logrm
Usage: ./logrm [-s] <file1> [<file2> ...]
  -s  Delete file(s) silently without writing to the log
  -h  Show this help message
```

remove file

```
vagrant@ubuntu2210:~/data/as1$ rm file1
vagrant@ubuntu2210:~/data/as1$ rm -s file2
```

remove dir

```
vagrant@ubuntu2210:~/data/as1$ rm dir1
vagrant@ubuntu2210:~/data/as1$ rm -s dir2
```

.remove_log

```
Sun Mar  5 09:33:27 PM UTC 2023: vagrant: Deleted file file1
Sun Mar  5 09:35:45 PM UTC 2023: vagrant: Deleted directory dir1 and its contents
```

Script three:

help message:

```
vagrant@ubuntu2210:~/data/as1$ ./lnnumb
Usage: ./lnnumb <file1> [<file2> ...]
  -h  Show this help message
```

using file1 and file2

```
vagrant@ubuntu2210:~/data/as1$ ./lnnumb file1 file2
```

returns

```
----------------------
File Name: file1
File Owner: vagrant
----------------------
1 : line 1
2 : line 2
3 : something
4 : this is a test
5 : for the
6 : lnnumb
7 : script
----------------------
File Name: file2
File Owner: vagrant
----------------------
1 : this
2 : is the
3 : other
4 : file
```

```
5 : that is
6 : test
```