**Tasks**

**Basics (mandatory)**

* Print usage
* List tasks
  + A todo task has (at least) a completed state and a description
* Empty list
* Add new task
* Check task
* Remove task
* Argument error handling

**Advanced (optional)**

* Add new task error handling
* Remove task error handling
* Check task error handling
* Write unit tests for any unit it feels possible
* Refactor the application to align with the proposed architecture
* Anything else that comes to your mind
  + longhand commands
  + list only undone tasks with -l and all tasks with -la
  + add/remove/check more items with one command
  + Multiple users

**Print usage**

* **Given** the terminal opened in the project directory
* **When** the application is ran without any arguments
* **Then** it should print the usage information

$ todo

Command Line Todo application

=============================

Command line arguments:

    -l   Lists all the tasks

    -a   Adds a new task

    -r   Removes an task

    -c   Completes an task

**List tasks**

* **Given** the terminal opened in the project directory
* And the file where you store your data
* And a task with the description Walk the dog stored in the file
* And a task with the description Buy milk stored in the file
* And a task with the description Do homework stored in the file
* **When** the application is ran with -l argument
* **Then** it should print the tasks that are stored in the file
* And it should add numbers before each

$ todo -l

1 - Walk the dog

2 - Buy milk

3 - Do homework

**Empty list**

* **Given** the terminal opened in the project directory
* And the file where you store your data
* And the file has 0 task
* **When** the application is ran with -l argument
* **Then** it should show a message like this: No todos for today! :)

**Add new task**

* **Given** the terminal opened in the project directory
* **When** the application is ran with the -a "Feed the monkey" argument
* **Then** it should add a new task with the description *Feed the monkey*

**Add new task error handling**

* **Given** the terminal opened in the project directory
* **When** the application is ran with the -a argument
* **Then** it should show an error message like: Unable to add: no task provided

**Remove task**

* **Given** the terminal opened in the project directory
* And the file where you store your data
* And the file has at least 2 tasks
* **When** the application is ran with the -r 2 argument
* **Then** it should remove the second task from the file

**Remove task error handling**

* 1.
  + **Given** the terminal opened in the project directory
  + **When** the application is ran with the -r argument
  + **Then** it should show an error message like: Unable to remove: no index provided
* 2.
  + **Given** the terminal opened in the project directory
  + And the file where you store your data
  + And the file has less than 20 tasks
  + **When** the application is ran with the -r 20 argument
  + **Then** it should show an error message like: Unable to remove: index is out of bound
* 3.
  + **Given** the terminal opened in the project directory
  + **When** the application is ran with the -r apple argument
  + **Then** it should show an error message like: Unable to remove: index is not a number

**Argument error handling**

* **Given** the terminal opened in the project directory
* **When** the application is ran with an unsupported argument *(eg. get)*
* **Then** it should show an error message like: Unsupported argument
* And it should print the usage information

**Check task**

* **Given** the terminal opened in the project directory
* And the file where you store your data
* And the file has at least 2 tasks
* **When** the application is ran with the -c 2 argument
* **Then** it should check the second task from the file

**Print all tasks**

* **Given** the terminal opened in the project directory
* And a undone task with the description Walk the dog stored in the file
* And a done task with the description Buy milk stored in the file
* And a undone task with the description Do homework stored in the file
* **When** the application is ran with -l argument
* **Then** it should print the tasks that are stored in the file
* And it should add [ ] before each if its undone otherwise [x]

1 - [ ] Walk the dog

2 - [x] Buy milk

3 - [ ] Do homework

**Check task error handling**

**Missing Index**

* **Given** the terminal opened in the project directory
* **When** the application is ran with the -c argument
* **Then** it should show an error message like: Unable to check: no index provided

**Index is not found**

* **Given** the terminal opened in the project directory
* And the file where you store your data
* And the file has less than 20 tasks
* **When** the application is ran with the -c 20 argument
* **Then** it should show an error message like: Unable to check: index is out of bound

**Invalid argument type**

* **Given** the terminal opened in the project directory
* **When** the application is ran with the -c apple argument
* **Then** it should show an error message like: Unable to check: index is not a number