# Todo App

## Goal

Write an application that we can use to manipulate our "tasks.txt" file. By executing it with various parameters we can manipulate our file without opening it.

Example tasks.txt

```
[ ] Buy milk
[x] Buy eggs
[ ] Buy bread
[x] Buy cheese
```

The application will **not** be interactive, instead, it parses the arguments being passed through the command line which will then result in manipulating the file.

The format of the tasks.txt is simple:

- Every line is a task
- If a task starts with "`[ ]`" then it is considered an incomplete task
- If a task starts with "`[x]`" then it is considered a complete task

# Steps

## Step 1: `todo list done`

This will read the "tasks.txt" and print only the tasks that are done. It will also output the number of tasks that have been completed.

```
$ ./todo list done
2 tasks done
[x] Buy eggs
[x] Buy cheese
```

## Step 2: `todo list todo`

This will read the "tasks.txt" and print only the tasks that are done. It will also output the number of tasks that have been completed.

```
$ ./todo list todo
2 tasks todo
[ ] Buy milk
[ ] Buy bread
```

## Step 3: `todo list all`

This will read the "tasks.txt" and print all tasks to the screen. It will also output the number of tasks that are complete and incomplete.

```
$ ./todo list all
2 tasks todo
[ ] Buy milk
[ ] Buy bread
2 tasks done
[x] Buy eggs
[x] Buy cheese
```

## Step 4: `todo add "Eat lunch"`

This will add "`[ ] Eat lunch`" to "tasks.txt" and print "Task has been added" to the screen.

```
$ ./todo add "Eat lunch"
Task has been added
```

## Step 5: `todo done "Buy milk"`

This will search the file "tasks.txt" for "`[ ] Buy milk`" and replace it with "`[x] Buy milk`".

```
$ ./todo done "Buy milk"
Task has been marked as completed
```

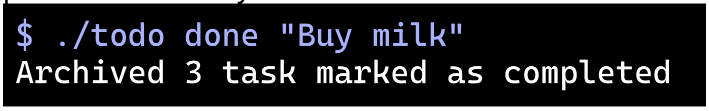
## Step 6: `todo undo "Buy milk"`

This will search the file "tasks.txt" for "`[x] Buy milk`" and replace it with "`[ ] Buy milk`".

```
$ ./todo undo "Buy milk"
Task has been marked as incomplete
```

## Step 7: `todo archive`

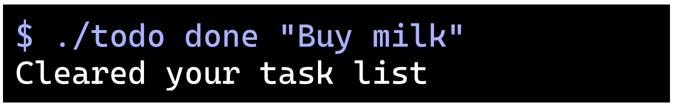
This will search the file "tasks.txt" for any line starting with "[x] " and remove it. In addition, it will print out how many tasks have been archived.



```
$ ./todo done "Buy milk"
Archived 3 task marked as completed
```

## Step 8: `todo clear`

This will clear the file "tasks.txt"



```
$ ./todo done "Buy milk"
Cleared your task list
```

## Bonus steps

Only perform these if you have completed the other tasks.

1. If "tasks.txt" doesn't exist ask the user if it should create one with dummy entries.
2. When archiving move the tasks to another file instead of just removing it.
3. When clearing ask if the user is certain
4. Check/warn if the task already exists when adding it as done or as complete

## Starter template

```
if (args.Any())
{
    // Remove executable if called from external path
    if(args[0].EndsWith(".dll") || args[0].EndsWith(".exe")) {
        args = args[1..];
    }
}

// Console.WriteLine("App was launched with the following arguments:");
// foreach (var arg in args)
// {
//     Console.WriteLine(arg);
// }
```