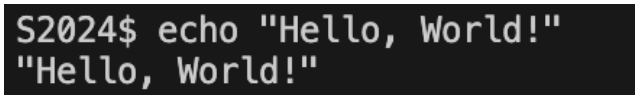
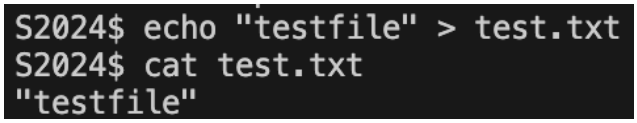
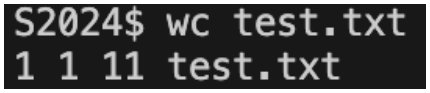
The reason my test data below is valid is because I use commands that are going to show the desired output of the code. I show how my code is capable of executing the basic echo, cat, and wc commands. I also show how my code is capable of utilizing parenthesis and semicolons to execute the commands in its desired priority. My testing data is valid because it is capable of showing all the capabilities of this code while proving the outputs are in check with the requirements of this assignment.

1. After your shell has started, it will give a prompt to the user. You should use “S2024$” for the prompt.

2. Make sure that your environment can run commands such as “echo”, “cat” or “wc” so that you can test your results. Some of your commands will take arguments.







3. After the prompt, the user can enter a command line consisting of commands and the “;” and “()” operators.

4. The “;” operator establishes the execution order of the commands from left to right. For example, if the user types `cmd1 arg;cmd2 arg2` after the prompt, `cmd1` will get executed first. `cmd2` will execute after `cmd1` terminates. After `cmd2` is executed, your shell will give a new prompt in a new line, ready to execute the next command line.

```
S2024$ echo "First command"; echo "Second command"
"First command"
"Second command"
```

5. The operators are processed from left to right. The parenthesis operators may change the execution order of the commands. Commands enclosed in a pair of parenthesis executes ahead of those outside of the parenthesis. For example, `cmd1;(cmd2;cmd3)` will see `cmd2` executes first, `cmd3` executes second and `cmd1` executes last.

```
S2024$ (echo "First"; (echo "Second"; echo "Third"))
"Second"
"Third"
"First"
```

6. Your shell shall support a command line with a string of 5 or more commands connected by the ";" and parenthesis operators with up to 2 levels of nesting.

```
S2024$ echo "One"; echo "Two"; echo "Three"; echo "Four"; echo "Five"
"One"
"Two"
"Three"
"Four"
"Five"
```

7. Your shell will identify illegal command lines. That is, command lines that cannot be executed by your shell. What are they?

   Illegal command lines are errors in the syntax. When we misuse a semicolon or forget to close a parenthesis these are the illegal command lines. Below is an example of how they are handled by my code.

```
S2024$ echo ; "mist"
exec "mist" failed
```

```
S2024$ (echo "Mismatched"
invalid commandSyntax
init: starting sh
```