Quiz: POSIX Shell III (Practice Problems)

1 Glob

Note 1. The POSIX shell has a built-in pattern matching feature for working with files. The glob operator `*` matches zero or more of any character, and the question operator `?` matches zero or one of any character. The `*` and `?` operators do not match a dot at the beginning of the file, and so do not match hidden files.

Problem 1. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch hello world
3 $ touch hola mundo
4 $ touch salve munde
5 $ rm *e*
6 $ ls | wc -l
```

Problem 2. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch hello world
3 $ touch hola mundo
4 $ touch salve munde
5 $ rm e*
6 $ ls | wc -l
```

Problem 3. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch hello world
3 $ touch hola mundo
4 $ touch salve munde
5 $ rm *e
6 $ ls | wc -l
```

Problem 4. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch .hello world
3 $ touch .hola mundo
4 $ touch .salve munde
5 $ rm *e*
6 $ ls -a | wc -l
```

Problem 5. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch .hello world
3 $ touch .hola mundo
4 $ touch .salve munde
5 $ rm .*e
6 $ ls -a | wc -l
```

Problem 6. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch "hello world"
3 $ touch "hola mundo"
4 $ touch "salve munde"
5 $ rm *d?
6 $ ls | wc -l
```

Problem 7. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch "hello world"
3 $ touch "hola mundo"
4 $ touch "salve munde"
5 $ rm *d?
6 $ ls | wc -l
```

1.1 Weirdness

Note 2. The glob does not expand within quotes. If the glob expression has no matches, then the literal expression is passed as an argument.

Problem 8. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch "hello world"
3 $ touch "hola mundo"
4 $ touch "salve munde"
5 $ touch *
6 $ ls | wc -l
```

Problem 9. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch *
3 $ ls | wc -l
```

Problem 10. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch *
3 $ ls | wc -l
```

Problem 11. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch "hello world"
3 $ touch "hola mundo"
4 $ touch "salve munde"
5 $ touch "*"
6 $ ls | wc -l
```

Problem 12. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch "hello world"
3 $ touch "hola mundo"
4 $ touch "salve munde"
5 $ touch "*"
6 $ ls | wc -l
```

1.2 For loops

Note 3. Glob expansion happens after the shell processes the spaces that separate the list of strings to loop over.

Problem 13. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch "hello world"
3 $ touch "hola mundo"
4 $ touch "salve munde"
5 $ for i in *; do echo $i; done | wc -l
```

Problem 14. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch hello world
3 $ touch hola mundo
4 $ touch salve munde
5 $ for i in *; do echo $i; done | wc -l
```

Problem 15. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ touch hello world
3 $ touch hola mundo
4 $ touch salve munde
5 $ for i in "*"; do echo $i; done | wc -l
```

Problem 16. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ for i in *; do echo $i; done | wc -l
```

1.3 Security

Note 4. Glob expansion happens in the shell, before the parameters are sent to the program. This can have unintended side effects. If you are working in a directory where someone else is allowed to create files, they can create files that will be expanded by `*` into command line arguments. This problem can be mitigated by using `./*` instead of `*`. Command line arguments that appear after a `--` will never be interpreted as command line arguments.

Problem 17. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir test
3 $ rm *
4 $ ls
```

Problem 18. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir test
3 $ echo evil > -rf
4 $ rm *
5 $ ls
```

Problem 19. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir test
3 $ echo evil > -rf
4 $ rm ./*
5 $ ls
```

Problem 20. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir test
3 $ rm -- -rf *
4 $ ls
```

Problem 21. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir test
3 $ rm -rf -- *
4 $ ls
```

Problem 22. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir -- -a
3 $ echo evil > -a/evil
4 $ ls *
```

Problem 23. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ mkdir -- -a
3 $ echo evil > -a/evil
4 $ ls -- *
```

2 Fun with git and glob

Problem 24. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ git init
$ touch hello world
$ touch .salve .munde
$ git add *e*
$ git commit -m 'first commit'
$ git checkout -b foo
$ git add *
$ git commit -m 'second commit'
$ git checkout master
$ git checkout -b bar
$ git add .
$ git commit -m 'third commit'
$ git checkout master
$ ls -a
```

Problem 25. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ git init
$ touch hello world
$ touch .salve .munde
$ git add .
$ git commit -m 'first commit'
$ git checkout -b foo
$ touch '*'
$ git add *
$ git commit -m 'second commit'
$ git checkout master
$ git checkout -b bar
$ echo "help me" > test
$ git add *
$ git commit -m 'third commit'
$ git checkout foo
$ ls -a
```

Problem 26. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ git init
$ mkdir test
$ touch test/hello world
$ touch test/.salve .munde
$ cd test
$ git add .*
$ git commit -m 'first commit'
$ git checkout -b foo
$ git add .
$ git commit -m 'second commit'
$ git checkout master
$ ls -a
```

Problem 27. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ git init
$ mkdir test
$ touch hola mundo
$ touch test/'hello world'
$ touch test/'.salve .munde'
$ cd test
$ for i in *; do git add $i; done
$ git commit -m 'first commit'
$ git checkout -b foo
$ git add .
$ git commit -m 'second commit'
$ ls -a
```

Problem 28. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ git init
$ echo evil > -a
$ touch hola mundo
$ touch test/'hello world'
$ touch test/'.salve .munde'
$ cd test
$ git add .
$ git commit -m 'first commit'
$ git checkout -b foo
$ git add ..
$ git commit -m 'second commit'
$ cd $HOME/quiz
$ git checkout master
$ ls *
```