

## 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 exactly 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ git init
3 $ touch hello world
4 $ touch .salve .munde
5 $ git add *e*
6 $ git commit -m 'first commit'
7 $ git checkout -b foo
8 $ git add *
9 $ git commit -m 'second commit'
10 $ git checkout master
11 $ git checkout -b bar
12 $ git add .
13 $ git commit -m 'third commit'
14 $ git checkout master
15 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ git init
3 $ touch hello world
4 $ touch .salve .munde
5 $ git add .
6 $ git commit -m 'first commit'
7 $ git checkout -b foo
8 $ touch '*'
9 $ git add *
10 $ git commit -m 'second commit'
11 $ git checkout master
12 $ git checkout -b bar
13 $ echo "help me" > test
14 $ git add *
15 $ git commit -m 'third commit'
16 $ git checkout foo
17 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ git init
3 $ mkdir test
4 $ touch test/hello world
5 $ touch test/.salve .munde
6 $ cd test
7 $ git add .*
8 $ git commit -m 'first commit'
9 $ git checkout -b foo
10 $ git add .
11 $ git commit -m 'second commit'
12 $ git checkout master
13 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ git init
3 $ mkdir test
4 $ touch hola mundo
5 $ touch test/'hello world'
6 $ touch test/' .salve .munde'
7 $ cd test
8 $ for i in *; do git add $i; done
9 $ git commit -m 'first commit'
10 $ git checkout -b foo
11 $ git add .
12 $ git commit -m 'second commit'
13 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ git init
3 $ echo evil > -a
4 $ touch hola mundo
5 $ touch test/'hello world'
6 $ touch test/' .salve .munde'
7 $ cd test
8 $ git add .
9 $ git commit -m 'first commit'
10 $ git checkout -b foo
11 $ git add ..
12 $ git commit -m 'second commit'
13 $ cd $HOME/quiz
14 $ git checkout master
15 $ ls *
```