Shell Topic 05: The Glob

Note 1. The POSIX shell has a built-in pattern matching feature for working with files. This is one of the most powerful features of the shell, but also one of the most dangerous. The *glob* operator `*` matches zero or more of any character, the *wildcard* operator `?` matches exactly one of any character, and the *character class* operators `[ ]` match any character contained within the square brackets. There are of course some subtleties:

1. The behavior of `*` and `?` in the shell is related to the bahavior in regex, but not exactly the same. In regex, these two operators modify the previous expression, in the shell they are a command by themselves and do not modify anything. The shell glob and wildcard operators were invented first in the 1960s and the regex syntax was developed in the 70s and 80s.
2. If the first character between the square brackets is `^`, then the character class matches any character not contained with in the brackets. This behavior is the same between shell and regex.
3. The glob and wildcard operators do not match a dot at the beginning of the file, and so do not match hidden files.

Problem 2. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch hello world
$ touch hola mundo
$ touch salve munde
$ rm *e*
$ ls | wc -l
```

Fraction of LLMs with correct answer: 6 / 13 = 0.46

Problem 3. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch hello world
$ touch hola mundo
$ touch salve munde
$ rm e*
$ ls | wc -l
```

Fraction of LLMs with correct answer: 6 / 13 = 0.46

Problem 4. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch hello world
$ touch hola mundo
$ touch salve munde
$ rm *e
$ ls | wc -l
```

Fraction of LLMs with correct answer: 8 / 13 = 0.62

Problem 5. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch hello world
$ touch hola mundo
$ touch salve munde
$ rm *[eo]
$ ls | wc -l
```

Fraction of LLMs with correct answer: 9 / 13 = 0.69

Problem 6. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch hello world
$ touch hola mundo
$ touch salve munde
$ rm *[^eo]
$ ls | wc -l
```

Fraction of LLMs with correct answer: 5 / 13 = 0.38

Problem 7. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 6 / 13 = 0.46

Problem 8. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 6 / 13 = 0.46

Problem 9. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 10 / 13 = 0.77

Note 10. The glob does not expand within quotes. If the glob expression has no matches, then the literal expression is passed as an argument.

Problem 11. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch "hello world"
$ touch "hola mundo"
$ touch "salve munde"
$ touch *
$ ls | wc -l
```

Fraction of LLMs with correct answer: 8 / 13 = 0.62

Problem 12. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 8 / 13 = 0.62

Problem 13. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ touch "hello world"
$ touch "hola mundo"
$ touch "salve munde"
$ touch "*"
$ ls | wc -l
```

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Note 14. The glob implicitly places quotation marks around all of the file names it expands into, so tokenization happens "the right way".

Problem 15. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 16. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Problem 17. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 11 / 13 = 0.85

Problem 18. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 6 / 13 = 0.46

Note 19. Glob expansion happens in the shell, before the parameters are sent to the program. This can have unintended security-related 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:

1. Using `./*` instead of `*`. The former will prepend `./` to all filenames, and so these files are unlikely to be interpretted as command line arguments.
2. Using `--` before any glob operations. It is customary for commands to interpret all arguments after `--` as filenames, and so it is safe to use the glob after `--`.

Problem 20. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ mkdir test
$ rm *
$ ls
```

Fraction of LLMs with correct answer: 9 / 13 = 0.69

Problem 21. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ mkdir test
$ echo evil > -rf
$ rm *
$ ls
```

Fraction of LLMs with correct answer: 7 / 13 = 0.54

Problem 22. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ mkdir test
$ echo evil > -rf
$ rm ./*
$ ls
```

Fraction of LLMs with correct answer: 8 / 13 = 0.62

Problem 23. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ mkdir test
$ rm -- -rf *
$ ls
```

Fraction of LLMs with correct answer: 12 / 13 = 0.92

Problem 24. Write the output of the final command in the following shell script.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ mkdir test
$ rm -rf -- *
$ ls
```

Fraction of LLMs with correct answer: 13 / 13 = 1.00

Problem 25. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 0 / 13 = 0.00

Problem 26. Write the output of the final command in the following shell script.

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

Fraction of LLMs with correct answer: 5 / 13 = 0.38

LLM Model Performance