

**NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.**

**SCENARIO:**

You are developing a library for filesystem operations. One of the tasks is to implement a recursive version of the clean command, which accepts a directory path and deletes its contents recursively if required. In the following implementation of such feature, the `os.fwalk` function generates file names in a given directory tree by walking the tree either top-down or bottom-up. The `os.unlink` and `os.rmdir` functions remove (i.e. delete) a file path and a directory path respectively. Consider the snippet of code below and answer the following questions, assuming that the code has all required permissions to execute.

```
01  import os
02
03  def clean(path):
04      for root, dirs, files, root_fd in os.fwalk(path, topdown=False):
05          for name in files:
06              os.unlink(name, dir_fd=root_fd)
07          for name in dirs:
08              os.rmdir(name, dir_fd=root_fd)
```

Questions:

1. What will the `clean` function do once it is executed?
2. What will happen if the directory passed to the function exists and contains symbolic links pointing to some files?
  - a. The function deletes only path and everything inside it recursively if it is a directory, including symbolic links (but not files they are pointing to), and exits.
  - b. The function throws an exception with a message because of the existence of symbolic links.
  - c. The function deletes files to which the symbolic links are pointing, not the symbolic links themselves.
  - d. None of the above.

*[Other statistical questions will be imported here while creating survey.]*

**NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.**

**ANSWER:**

1. Program will take a string as a path and delete it, and its content if the path is directory.

2. a

The default value for `follow_symlinks` is `False`, which means the operation is done on symbolic link itself, not where it points.

**TAGS:**

python, flawless

**CATEGORIES:**

Blindspot - No

Type - File

Number of distinct functions - 3

Number of total functions - 3

Blindspot function - N/A

Function call omitted - N/A

Blindspot type - N/A

Number of Parameters in the blindspot function - N/A

Cyclomatic complexity - 2

**NAME:**

`os.fwalk()`

**DESCRIPTION:**

The `os.fwalk` behaves exactly like `os.walk`, except that it yields a 4-tuple (`dirpath`, `dirnames`, `filenames`, `dirfd`), and it supports `dir_fd`. `dirpath`, `dirnames` and `filenames` are identical to `os.walk` output, and `dirfd` is a file descriptor referring to the directory `dirpath`.

**BLINDSPOT:**

#N/A

**CORRECT USE EXAMPLE:**

#N/A

**\$MORE INFORMATION:**

#N/A

**REFERENCES:**

1. <https://docs.python.org/3/library/os.html?highlight=walk#os.fwalk>

