

## ENGR 221 — Lab 6 answers

Student: Joe Le

ENGR 221 — Fall 2025

Assignment: Lab 6 (Random Walk & Recursion)

### Part 1 — Understanding randomWalker.py

#### 1.b — Function rs()

i) What this function does:

rs() returns a single random step for the sleepwalking student. Each call chooses either -1 (a step to the left) or +1 (a step to the right) with equal probability.

ii) What value(s) does it return:

It returns an integer value of either -1 or +1.

iii) Is that value always the same:

No. Each call to rs() is random and independent, so the result can vary.

#### 2.b — Function rwpos(start, low, hi)

i) What this function does:

rwpos() recursively simulates the student taking random steps until they reach or cross an endpoint. It returns the boundary value where the student stops (either low or hi).

ii) What is the base case:

If start <= low, return low; if start >= hi, return hi.

iii) What values are passed to the recursive call:

A new starting position (start + rs()), along with the same low and hi bounds.

iv) What value is returned by the original function call:

The final position where the student stops (either low or hi).

#### 3.b — Function rwsteps(start, low, hi)

i) What this function does:

rwsteps() prints the hallway with the student's position marked by 'S', then recursively calls itself until the student reaches or passes an endpoint. It returns the total number of steps taken to reach that endpoint.

ii) What is the base case:

If start <= low or start >= hi, return 0 (no more steps to take).

iii) What values are passed to the recursive call:

A new position (start + rs()) with the same low and hi bounds.

iv) What is printed each time `rwsteps()` is called:

A hallway visualization where 'S' marks the student's current position, followed by the values 'start low hi' separated by fixed spacing.

v) What value is returned by the original function call:

The total number of steps taken to reach the boundary (a non-negative integer).

## Part 2 — `rwstepsLoop()`

In Part 2, `rwstepsLoop()` was implemented as a loop-based equivalent of `rwsteps()`. It performs the same logic without recursion — taking random steps, printing the hallway each time, and stopping when the student reaches either boundary. It returns the number of steps taken. The function passes all tests in `walker_tests.py`.

## Part 3 — Testing and GitHub Submission

Commands to test and submit:

1. Run tests from your Lab6 folder:

```
pytest -v tests/walker_tests.py
```

(or on Windows: `py -m pytest -v .\tests\walker_tests.py`)

2. Push changes to GitHub:

```
git add --all
```

```
git commit -m 'Lab 6 commit'
```

```
git push
```

3. Submit this .docx file as your Lab 6 answers on Canvas.