# random\_walk\_1d.py  
from random import choice  
from typing import List, Tuple  
  
  
def step\_1d() -> int:  
 *"""Return a single step: -1 (left) or +1 (right)."""* return choice([-1, 1])  
  
  
def walk\_1d(n\_steps: int) -> List[int]:  
 *"""  
 Simulate a 1-D random walk for n\_steps.  
 Returns the list of positions after each step (length == n\_steps).  
 Start at position 0 (not included in the list).  
 """* pos = 0  
 path: List[int] = []  
 for \_ in range(n\_steps):  
 pos += step\_1d()  
 path.append(pos)  
 return path  
  
  
def stats\_1d(path: List[int]) -> Tuple[int, int, int, int]:  
 *"""  
 Return (final\_pos, min\_pos, max\_pos, returns\_to\_origin).  
 A 'return to origin' counts each time the path position equals 0.  
 """* final\_pos = path[-1] if path else 0  
 min\_pos = min(path) if path else 0  
 max\_pos = max(path) if path else 0  
 returns = sum(1 for p in path if p == 0)  
 return final\_pos, min\_pos, max\_pos, returns  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 N = 1000 # try 10, 100, 1000  
 path = walk\_1d(N)  
 final\_pos, min\_pos, max\_pos, returns = stats\_1d(path)  
  
 print(f"1-D walk for {N} steps")  
 print(f"Final position: {final\_pos}")  
 print(f"Min/Max visited: {min\_pos}/{max\_pos}")  
 print(f"Returns to origin: {returns}")