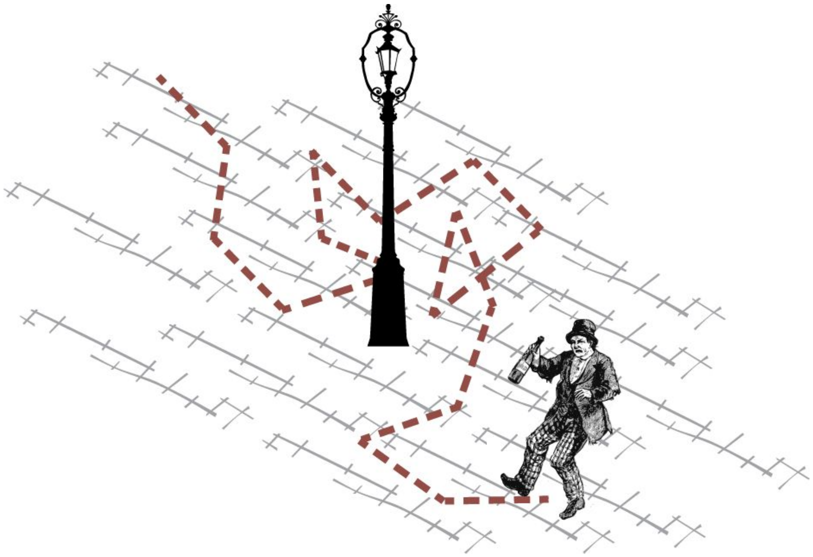
Array HW 2 - Random Walk



Artificial intelligence is based on evaluating past decisions and predicting (hopefully with a high degree of certainty) what future events will be. The first step to evaluating past decisions is to store data and evaluate trends. This assignment, based on arrays and random numbers, will explore the first step in this evaluation process.

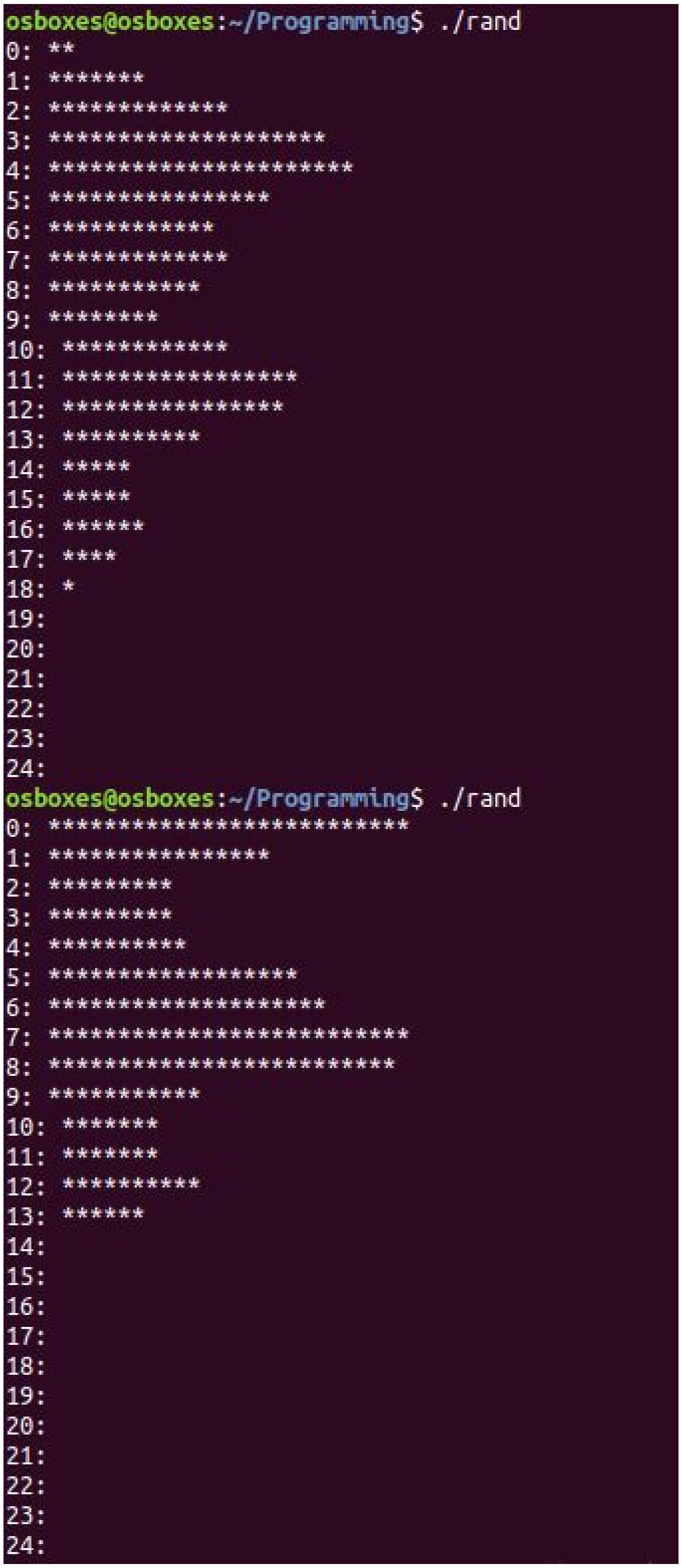
The idea behind this assignment is to model the behavior of a random walk (also known as the "drunkard's walk) using 1D arrays. Using an array of 25 spaces long, all with a beginning value of 0, the "user" will start in the middle (the 12th element). If a random number is 0, then the user will move down one element (subtract 1). If the random number is 1, then the user will move up one element (add 1). Every time that the user reaches an space (including the starting element), the user will add one to the value of that element.

After 100 rounds, the user should print out the array - each element will have 1 line, starting at 0 and increasing to 25. Each line should print the number of *s associated with the value stored in each element.

Notes

- Do not let the array go "out-of-bounds" - that is, if the user is at [0] and rolls a 0, the user should stay at [0].
- Each round of 100 will display a slightly different map - if this is the case, your code is working correctly. There is no one correct map.

Coding example: Final Code



```
osboxes@osboxes:~/Programming$ ./rand
0: **
1: *******
2: *************
3: ********************
4: **********************
5: ****************
6: ************
7: *************
8: ***********
9: ********
10: ************
11: *****************
12: ****************
13: **********
14: *****
15: *****
16: ******
17: ****
18: *
19:
20:
21:
22:
23:
24:
```

In the first code, the random walk traveled slightly in a positive direction (as evidenced by the occurrences in element 13 and beyond), but mainly traveled in a negative direction.

```
osboxes@osboxes:~/Programming$ ./rand
0: **************************
1: ****************
2: *********
3: *********
4: **********
5: ******************
6: ********************
7: **************************
8: *************************
9: ***********
10: *******
11: *******
12: **********
13: ******
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24:
```

Similar to the first, this random walk traveled mainly in the negative direction (only moving beyond the middle 6 times), and appeared to be stuck at the far negative end.