THE RANDOM SLEEPWALKER

Imagine that a student is stuck at the fountains in the center of CSI!

Due to total exhaustion (exams, papers, projects, homework, programs!), the student falls asleep and begins walking.

The student takes a random step either forward (toward the library) or backward (toward the student Center). The student keeps taking these random steps until the student hits one of these buildings.

Write a program that simulates this student’s walk (this is called a ‘Random Walk’). See an example .exe from the CSC126 page. If you cannot run it from your browser, ask your lab instructor to demo this on your instructor’s machine. A sample output is at the bottom of this page.

- Your program asks the user to enter a left and right border number (for example 0 and 10) if there are 10 steps between the two buildings. Your program asks for the students position (for example 5, if the student is halfway between the two buildings).
- For each step the student takes, your program calls a function that randomly returns either a 1 or a -1 to indicate which direction the step is. (this is a value returning function!) Remember srand(time(0)) at the start of your program. Then call rand() to give two choices!
- For each step the student takes, your program calls a function that prints the diagram with the current position (this is a void function!).
- The student stops when he/she hits one of the buildings.
- PRINT HOW MANY STEPS THE STUDENT HAD TO TAKE TO REACH A SIDE.
- SIMULATION: RUN YOUR PROGRAM 10 times with the input 0  10  5 Record the average number of steps it took to arrive at one of the buildings in comments at the top of your code!

Sample OUTPUT:

Enter the left edge and right edge of the path: For example 0  10
0 10

Enter the sleepwalking student's current position:

For example (if the student is halfway between put in a 5
5
-----$-----
-----$-----
-----$-----
-----$-----
It took: 9 steps!

Enter any character to exit:

NOTE! In my simulation, my #includes are as follows:

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <thread>
using namespace std;
using namespace std::this_thread; // sleep_for, sleep_until
using namespace std::chrono; // nanoseconds, system_clock, seconds
```

and after each step I placed a wait call:

```cpp
sleep_for(microseconds(30000)); // waits so it does not fly by the screen
```

taken from: https://www.engage-csedu.org/find-resources/sleepwalking-student