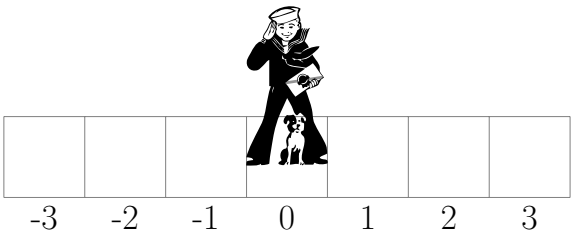
CSC 112 Lab 3
Drunken Stagger
Spring 2015

Due
3:00pm Friday, 2/13

1 Introduction

Consider a drunken sailor[1] staggering about. Assume the sailor is initially located in the middle of a board (plank) and can step either to the left or right. Given the board has a certain length and the sailor can take a maximum number of steps, what is the probability he will fall off the board?



1.1 The Odds of Falling

We are interested in determining the probability that the sailor will stagger (fall) off the board. A simple method for determining the probability is to have a sailor do this many times (each time is considered an experiment) and record the how often he falls. Of course, there is a limited supply of sailors in Winston-Salem area, so we'll rely on simulating a staggering person. As done in lab 2, the probability of falling would be the number of times the sailor falls divided by the total number of experiments performed. The more experiments that are performed, the better the estimation.

The program will prompt the user for the board length (measured in steps), the maximum number of steps to take, and the number of experiments. Note, the board length entered will be half of the actual length minus one. Therefore if the user enters 3 for the board length, then the actual length is 7 (as seen above). After the information is entered, your program will then run the number of experiments requested. For each experiment, graphically display the steps of the sailor and the result. At the end of all the experiments, display the total number of experiments, the number times the sailor fell, and the fall percentage (probability estimation). An example of a board of length 3 (7 actual length), 5 maximum steps, and 2 experiments is given below. Note this is only an example, not necessarily what you should expect.

Terminal

```
> ./lab3
Enter the board length, max number of steps and number of experiments -> 3 5 2
========
__*___
_*____
*_____
_*____
__*___
Sailor SAFE :)
========
___*__
____*_
_____*
Sailor FELL :(
After 2 experiments, sailor fell 1 time, fall percentage was 0.5%
```

[1] Of course the sailor is of age; regardless, we certainly don't condone alcohol. Booooo alcohol! It only leads to staggering.

2 Simulating a Drunken Stagger

Assume the board is divided into squares, where each square is a location the sailor can step. However, remember the sailor can only move one square to the left or right per step. Assign each square a number, where the middle square is zero, left squares are decreasing numbers, and right squares are increasing numbers. This is seen in the figure on the first page.

The stagger of the sailor is simply randomly choosing a -1 or +1 per step and adding it to the current position of the sailor (number of the square the sailor currently occupies). Remember the sailor will always start a square zero (the middle of the board). If this number is ever less than the leftmost square, then the sailor has fallen off the left edge. If this value is ever greater than the rightmost square, then the sailor has fallen off the right edge. If the sailor has taken the maximum number of steps (specified by the user) and remains on the board, then he has not fallen.

2.1 Programming Requirements for this Assignment

For this assignment, your program must at least consist of the following functions.

- `readExperimentData` is a function that will prompt the user and read the board length (measured in steps), the maximum number of steps to take, and the number of experiments to make.
- `drunkenWalk` is a function that will simulate one complete experiment. The function should return if the sailor fell.
- `displayBoard` is a function that will *graphically* display the current location of the sailor. The function will print an underline (`'_'`) for each board square except for the square where the sailor is located. Print an asterisk (`'*'`) for the square where the sailor is located.
- `outputExperimentStats` is a function that will display the number of experiments, the number of falls, and the percentage of falls.
- `randStep` is a function that will simulate a random step. The following function definition will simulate a step by randomly generating the value -1 or +1, then returns the value using the function name.

```
int randStep()
{  return (rand()%2?-1:1);  }
```

  Where is the function `rand()` defined? It is defined in the `cstdlib` library, so you must add the compiler directive `#include <cstdlib>` to your program.

As a result, your `main` function will consist of variable declarations, function calls, and control structures (loops). No output or input should be done in the `main` function.

3 Programming Points

You **must** adhere to all of the following points to receive credit for this lab assignment.

1. Create a directory `Lab3` off of your `CSC112` directory to store your program in a file called `lab3.cpp`
2. Your program must be modular in design and consist of the five functions described in Section 2.1.
3. Your `main` function can only consist of variable declarations, function calls, and control structures (no input or output in the `main` function).
4. Your program must compile cleanly, no errors or warnings are allowed.
5. Pass variables appropriately. Only pass by reference when necessary.
6. Your program must adhere to documentation style and standards. Don't forget function headers and variable declarations.
7. **Turn-in** a print-out of your program source code (`lab3.cpp`). In addition, copy your program source code to your `Gade/Lab3` directory.