CSE141 Introduction to Programming (Fall'24)

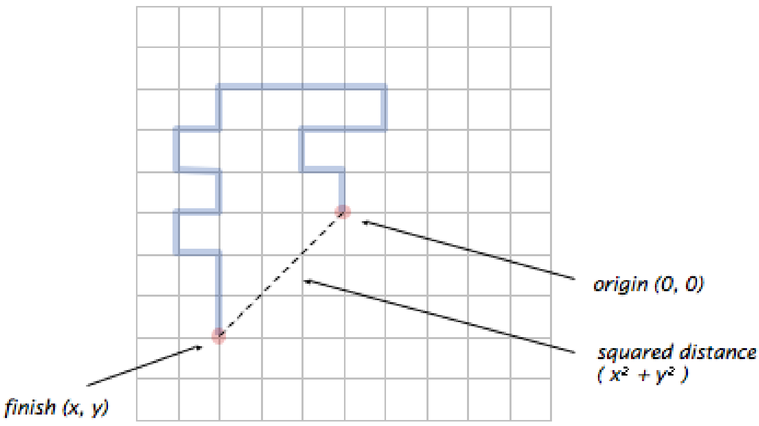
Homework #1 *Due:* Sep 22 by 11:55pm

Instrutions:

- Submit `num2eng.cpp`, `random_walker.cpp`, and `random_walkers.cpp` and write your name and ERP id on the first line in each as comment.
- *Late submission:* There will be 20% penalty for up to *one* day late submissions, 50% for *two* days late submissions. *No submission will be accepted after two days past the due date.*
- *Plagiarism:* Students are expected to perform their work individually unless otherwise specified by the instructor. Assignments may be discussed in general terms with other students and the students may receive assistance from the instructor, TA, or classmates. Assistance does not mean obtaining solutions and modifying them; this is considered plagiarism.

1. [15 marks] **Number-to-English.** Write a program `num2eng.cpp` to read input an integer between $-999,999$ and $999,999$ and print the English equivalent. Here is an exhaustive list of words that your program should use: negative, zero, one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety, hundred, thousand. Don't use hundred, when you can use thousand, e.g., use one thousand five hundred instead of fifteen hundred.

2. [15 marks] **A drone's flight.** A drone begins flying aimlessly, starting at IBA Student Center. At each time step, the drone flies one meter in a random direction, either north, east, south, or west, with probability 25%. How far will the drone be from Student Center after n steps? This process is known as a *two-dimensional random walk*[1].



[1] *This process is a discrete version of a natural phenomenon known as Brownian motion. It serves as a scientific model for an astonishing range of physical processes from the dispersion of ink flowing in water, to the formation of polymer chains in chemistry, to cascades of neurons firing in the brain.*

(a) Write a program `random_walker.cpp` that takes an integer n as input and simulates the motion of a random walk for n steps. Print the location at each step (including the starting point), treating the starting point as the origin $(0, 0)$. Also, print the square of the final Euclidean distance from the origin.

```
% random_walker.exe
Enter n: 10
(0, 0)
(0, -1)
(0, 0)
(0, 1)
(0, 2)
(-1, 2)
(-2, 2)
(-2, 1)
(-1, 1)
(-2, 1)
(-3, 1)
squared distance = 10
```

```
% random_walker.exe
Enter n: 20
(0, 0)
(0, 1)
(-1, 1)
(-1, 2)
(0, 2)
(1, 2)
(1, 3)
(0, 3)
(-1, 3)
(-2, 3)
(-3, 3)
(-3, 2)
(-4, 2)
(-4, 1)
(-3, 1)
(-3, 0)
(-4, 0)
(-4, -1)
(-3, -1)
(-3, -2)
(-3, -3)
squared distance = 18
```

(b) Write a program `random_walkers.cpp` that takes two inputs `n` and `trials`. In each of `trials` independent experiments, simulate a random walk of n steps and compute the squared distance. Output the mean squared distance (the average of the trials squared distances).

```
% random_walkers.exe
Enter n and trials: 100 10000
mean squared distance = 101.446

% random_walkers.exe
Enter n and trials: 100 10000
mean squared distance = 99.1674

% random_walkers.exe
Enter n and trials: 200 1000
mean squared distance = 195.75
```

```
% random_walkers.exe
Enter n and trials: 400 2000
mean squared distance = 383.12

% random_walkers.exe
Enter n and trials: 800 5000
mean squared distance = 811.8264

% random_walkers.exe
Enter n and trials: 1600 100000
mean squared distance = 1600.13064
```

As n increases, we expect the random walk to end up farther and farther away from the origin. But how much farther? Use `random_walkers.cpp` to formulate a hypothesis as to how the mean squared distance grows as a function of n. Use `trials = 100000` to get a sufficiently accurate estimate.