Lab #7
Introduction to Algorithms

You are given an incomplete program SelectionSort_Lab.cpp, containing a main program and a function.

Complete the Selection Sort function by adding the necessary code, according to the pseudocode given in the class.

1. Complete the main program so that you can test the Selection Sort algorithm. Create an array with 100 elements, all even numbers, with the elements in descending (reverse) order: i.e. 200, 198, 196,…,2 using a for loop. After the Sort is complete, print out all the elements to check that they are sorted.

2. Modify the main program so that it will set up an array with 100 elements which are random numbers. Use the random number generator `rand()` function for this. Use an assignment like `A[i] = rand() % 100 + 1` to set up the numbers, and print them to check that they are "*random*". Sort them and then print them out to check they are sorted. Note, you need to `#include <stdlib.h>` for the random function.

Extra Work if not completed last time*. You may need to do this in your own time. Modify the Binary Search program, so that it creates and prints out the Table of values* for *the array shown below. Test it for both the keys* ***14 and 30****. The array of elements is:*

`12 13 14 16 17 18 21 24 27 30.`

Tutorial 7.

For the following 2 arrays show the contents of the array after **each Pass** of the Bubble Sort (improved version) and Selection Sort algorithms.

(i) 9 6 8 3 1 5 4

(ii) 9 1 2 3 4 5

For the Bubble Sort, and an array of size **n** in reverse order (worst case), draw a graph showing the number of comparisons (y-axis) needed for n = 10, 20, 30, 40 (x-axis). Comment on the shape of the graph.