Lab #8
Introduction to Algorithms

In this Lab we will learn about the **Stack data structure**, and its associated functions.
To see how a stack works, go to my web page at www.comp.dit.ie/smulligan/dt2111.html and click on the link to "The stack applet".

The two main functions are:
Push – Put an item on top of the stack.
Pop – Remove and return the item on top of the stack.

Here **top** points to the location of the element on top of the stack.
In my implementation the variable **top** points to the **first free location** on top of the stack, i.e. one more than for the applet.

You are given an incomplete program StackArray_Lab.cpp, containing a main program and two functions to **push and pop** an item on/off the stack.

Complete the push function by adding two lines of code inside the if statement. These should:
1. Copy the item into the stack at location top.
2. Increment the top by one, to indicate where the new top is.

Compile and run the program, and check that the **push** and **pop** functions are working properly. Fill the stack and check that it doesn't overflow, i.e. you get an error message. Empty the stack, and also check that you get an error message, i.e can't pop an item from an empty stack.

Make a copy of the program, and then modify the **main program** so that it uses a "for loop" to add the numbers 1,2,3,…10 to the stack.
Then (with another loop) pop() all the numbers and print them out, and verify that they are 10,9,…,2,1.

Tutorial 8.

Two algorithms, Algorithm A and Algorithm B can be used to solve the same problem. If the complexities of each algorithm are as follows, find the values of n for which each algorithm should be used so the number of operations is a minimum.

Algorithm A	Algorithm B
n^2	$20*n$
$n^2/3$	$10*n$
$10*n$	$n*\log_2(n)$
$n^3/3$	$20*n^2$

Hint. Find the values of n so that both algorithms take the **same number** of operations, and then look at smaller values and bigger values.