Algorithms & Computation Lab Sheet 2 Templates

1. Write a C++ template function named swap that has two parameters of the same type. The function swaps the value of two data types, test with float and integer.
   a. Change the types to pass by reference
   b. Change the types to pass by address

2. Write a C++ template function named multiples so that it has three parameters sum, x, and n. The first two parameters will have the type represented by the function template type parameter WhatKind. n will always be int. The return type is void. All parameters are passed by value except for sum which is passed by reference. A Template Function created from multiples will compute...

   ```
   sum = 1 + x + 2x + 3x + ... + nx
   ```

   So for example, test you multiples template function with "sum = 1 + x + 2x + 3x" therefore make x = 1 and n = 3 and the sum should be output as 7. Change x = 2 and sum should be 13.

3. Write a C++ template function named init so that it has three parameters whose types are determined by the function template type parameters T1 and T2. The function header is shown below. init sets the value of the parameter start to a T2-type value of 1. init returns a T1-type value which is the sum of num1 and num2.

   ```
   T1 init (T1 num1, T1 num2, T2& start)
   ```

   So for example a call to init with int num1 = 2; int num2 = 3; double start = 2.2 will return 5 and start will be set to integer 1

4. Create a template function for the QuickSort algorithm (http://en.wikipedia.org/wiki/Quicksort)

   ```
   template<typename T>
   T* quicksort(T* array, int start, int end)
   ```