**Name: Islam Osama Nwishy**

**ID#: 900170200**

**Assignment 7**

**Design Document**

1. **PQ [Class]:**
   * **H\_type:** enum that defines the type of a heap (either minimum or maximum)
   * **Type:** template class to store the type of the PQ
   * **Public:**
     + **PQ [constructor]:** took 4 parameters, the type of the heap (min by default), the size of the heap (11 by default), the value of element 0 (NULL by default)
     + **~PQ [destructor]:** deletes the array that is the queue
     + **Insert ():** takes an item of whatever type the Q is, inserts it to the array, and adjusts the heap. Returns false if the queue is full and true elsewise
     + **Remove ():** removes the top element from the Q and returns it. Adjusts the heap after removing the element
     + **Isempty ():** returns true if the pointer to the last element is at 0
     + **Peak():** returns the top element without removing it from the Q
   * **Private:**
     + **heapType:** a variable of type h\_type that holds the type of the heap (min or max)
     + **a:** a pointer to the array that acts as the Queue
     + **MaxSize:** the maximum size of the queue
     + **N:** the index of the last that exists in the array
     + **ItemMinOrMax:** holds the minimum or maximum possible value in the Queue to be stored at index 0
     + **UpHeap ():** adjusts the heap upwards
     + **DownHeap ():** adjusts the heap downwards
2. **App**
   * **Data [Class]:**
     + **Public:**
       - **Dates:** a variable of type string that holds the date of a given exchange day
       - **ExchangeRate:** a variable of type double that holds the Exchange rate of a given day
       - **ExchangeRateVsAve:** a variable of type double that holds the Exchange rate of a given day – the average rate of the whole dataset
       - **Operator functions:** overloading operators to compare data types based on their ExchangeRateVsAve variable
   * **GetInput ():** takes the information from a given file and inputs it into 2 given arrays one of type string that will hold the dates and the other of type double that will hold the exchange rates
   * **GetAverage ():** takes an array of type double and returns the average sum of its elements
   * **PopulateQ ():** populates a given Priority Queue based on the data stored in 2 arrays one for the dates and one for the exchange rates. It also calculates and stores the change of each exchange rate from the average rate
   * **PopulateArray ():** populates a given Array based on the data stored in 2 arrays one for the dates and one for the exchange rates. It also calculates and stores the change of each exchange rate from the average rate
   * **MaxSubarray ():** a function that solves the Maximum Subsequance Problem that takes an array and finds the starting and ending point of a subarray that yield the maximum sum of exchange rates change from the average
   * **Start ():** a function that handles the interface of the program as it takes in commands in the form of a number and carries out the appropriate operations for it