****

**BLG 335E**

**ANALYSIS OF ALGORITHMS I**

CRN: 10825

**REPORT OF HOMEWORK #1**

Submission Date: 22.10.2013

**STUDENT NAME: TUĞRUL YATAĞAN**

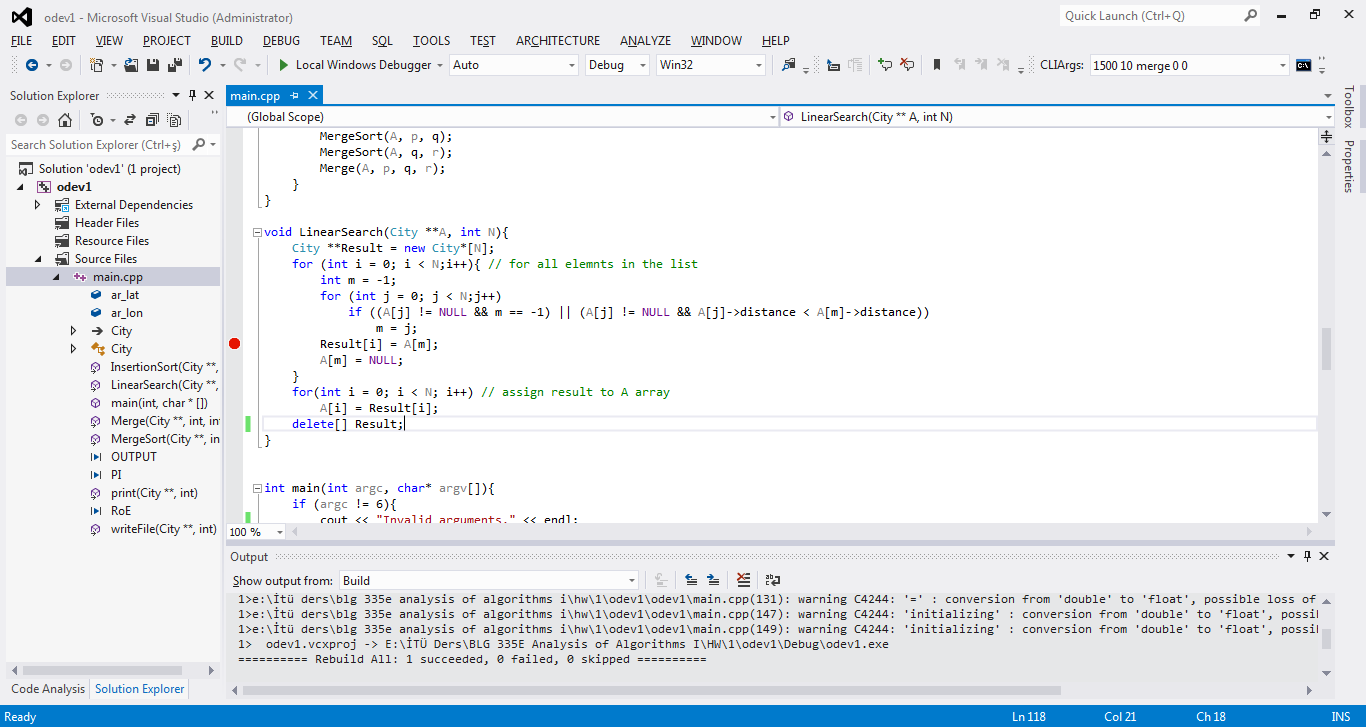
**STUDENT NUMBER: 040100117**

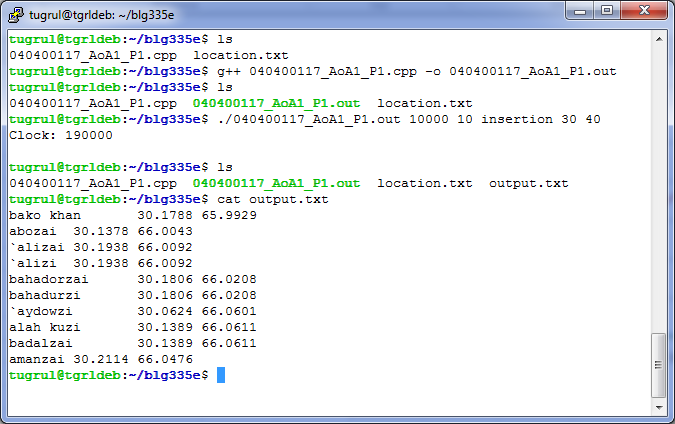
**1. Introduction**

In this project, we will find closest K locations among N locations to a given location considering their coordinates on earth.

**2. Development and Operating Environments**

Microsoft Visual C++ 2012 environment has been used to write the source code in Windows 7 operation system and GNU g++ compiler has been used for compiling under Debian 7 operation system.



The program built and compiled without any warning or error under g++. Finally the program is executed. Sample outcome is below:

**3. Data Structures and Variables**

The program takes 5 command line arguments. Example:

./040100117\_AoA1\_P1 **N K** **algorithmType** **latitude** **longitude**

**algorithmType** variables can be {“**insertion”**, “**merge”** or “**linear”**}

**N, K,** **latitude** and **longitude** variables can be integer value.

Example use of the program:

./040100117\_AoA\_P1 1000 10 insertion 30 40

**4. Analysis**

Running time of sorting functions according to K and N numbers are shown below in tables:







N is number of the element to be sorted and K is number of the element to demanded. As seen by the tables K does not effect on merge and insertion sorting, but K can effects only when linear sorting.

Merge sort is faster than insertion sort for N > 1000. After 1000, merge sort is becomes faster. Linear sort is handy only when K is very small or N is smaller than 1000.

Related graphs according to N and clock numbers for all three sorting algorithms separately for K numbers are shown below.

Note: For clarity of graph, I choose very large number (Ex:1000000) for ∞ when drawing graph. ∞ indicates that program takes very long time to execute or it cannot give any result in reasonable time.

We can investigate the graphs that; if we chose time functions of merge sort is m(n), insertion sort is i(n) and linear sort is l(n)

m(n) = O(l(n)) = O(i(n))

m(n) = O(l(n)) = O(i(n))

m(n) = O(l(n)) = O(i(n))

m(n) = O(i(n)) = O(l(n))

**5. Conclusion**

In this homework, I have become more familiar with the concept of analysis of algorithms. I had the chance to intensify my knowledge about instructing good and efficient algorithms.