**Code Challenge Instructions**Overview

This exercise asks you to write code that reads an input file, manipulates the contents, and produces one or more output files. There are two options to choose from below; pick one or both and write your solution in **C#**, **C++**, **VB**, **Java** or **Python**. You have until **5PM Central two days following** the receipt of these instructions to email us your source code.

This document contains confidential information that you may use only in connection with your employment application to kCura Corporation. The instructions, details and work product contained herein are protected under US and international copyright law, and are the exclusive property of kCura Corporation. No portion of this document, or the code that you develop from this exercise, may be distributed by any means to any person not employed by kCura Corporation.

Input File

Your solution must accept an input file via command line parameter. Each line of this input file will be formatted like so:

<Population, in hundreds of thousands>|<City>|<State>|<Semicolon-delimited list of interstates that run through this city>\n

Example:

4|Raleigh|North Carolina|I-40;I-85;I-95

27|Chicago|Illinois|I-94;I-90;I-88;I-57;I-55

10|San Jose|California|I-5;I-80

You may assume the following:

* The input file is well-formed: each pipe-delimited section will have one or more characters; the interstates section will have at least one interstate; the population number will be an integer; etc.
* All interstates have the “I-” prefix.
* The same city will not appear more than once in the input file.
* Chicago will be in the input file.

Accompanying these instructions is a file named **Sample\_Cities.txt** that you are encouraged to use to help construct your solution.

Option 1

Produce two output files from the input. The first must be named **Cities\_By\_Population.txt** and have data in the following format:

<Population>

(newline)

<City>, <State>

Interstates: <Comma-separated list of interstates, sorted by interstate number ascending>

(newline)

Cities must be ordered from highest population to lowest. If there are multiple cities with the same population, group them under a single <Population> heading and sort them alphabetically by state and then city.

Example output:

83

New York, New York

Interstates: I-78, I-80, I-87, I-95

27

Chicago, Illinois

Interstates: I-55, I-57, I-88, I-90, I-94

15

Phoenix, Arizona

Interstates: I-8, I-10, I-17

Philadelphia, Pennsylvania

Interstates: I-76, I-95

The second output file must be named **Interstates\_By\_City.txt** and contain a list of interstates and the number of cities they run through. Each line of the output file must be of the form:

<Interstate> <Number of cities>

Sort the list by interstate number ascending.

Example output:

I-5 5

I-10 4

I-19 1

I-20 3

Option 2

Produce a single output file named **Degrees\_From\_Chicago.txt**. Each line of the output file must be of the form:

<Degrees removed from Chicago> <City>, <State>

A city is considered 1 degree removed from Chicago if it shares an interstate with Chicago. A city that is not directly connected to Chicago but is to a city 1 degree removed is considered 2 degrees removed. And so on. Chicago itself is 0 degrees removed, and a city that is not directly or indirectly connected to Chicago has a degree of -1. Cities must only appear once, with the lowest degree of connection.

Sort the output by degree descending and then by city and state ascending.

Example output:

1 Boston, Massachusetts

1 Cleveland, Ohio

1 Seattle, Washington

0 Chicago, Illinois