Contestant Number: ________________

Time: _________

Rank: _________

JAVA PROGRAMMING
(340)

REGIONAL – 2016

Production Portion:

Program 1: Base K Addition	_______________	(335 points)
TOTAL POINTS	_____________	***(335 points)***

Failure to adhere to any of the following rules will result in disqualification:

1. **Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
2. **No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
3. **Electronic devices will be monitored according to ACT standards.**

No more than ten (10) minutes orientation
No more than 90 minutes testing time
No more than ten (10) minutes wrap-up

You will have 90 minutes to complete your work.

Your name and/or school name should NOT appear on any work you submit for grading.

Create a folder on the flash drive provided using your contestant number as the name of the folder. Copy your entire solution/project into this folder. You must submit your entire solution/project so that the graders may open your project to review the source code. You must ensure that the files required to run your program are present and will execute on the flash drive provided. Note that the flash drive letter may not be the same when the program is graded as it was when you created the program. It is recommended that you use relative paths rather than absolute paths to ensure that the program will run regardless of the flash drive letter. The graders will *not* compile or alter your source code to correct for this. Submissions that do *not* contain source code will *not* be graded.

Assumptions to make when taking this assessment:

- The input file will contain only ASCII characters.
- A test input file will be available and will be named, “basekaddition.txt”.

Development Standards:

- Your Code must use a consistent variable naming convention.
- All subroutines, functions, and methods must be documented with comments explaining the purpose of the method, the input parameters (if any), and the output (if any). If you create a class, then you must use Javadoc comments.

Your application will be graded on the following criteria:

Solution and Project

The project is present on the flash drive	____ 10 points
The projects main class is named baseKaddition	____ 10 points

Program Execution

The program runs from the USB flash drive	____ 15 points

If the program does *not* execute, then the remaining items in this section receive a score of zero.

The program runs and reads the input file	____ 20 points
The program displays an error message if the command line is empty	____ 20 points
The program displays an error message if the file cannot be found	____ 20 points
The program displays the base of the numbers	____ 20 points
The program displays the first number to add	____ 20 points
The program displays the second number to add	____ 20 points
The program displays the correct addition of the numbers	____ 30 points
The program displays all the output for all of the numbers in the file	____ 40 points

Source Code Review

The source code is properly commented	
A comment containing the contestant number is present	____ 10 points
Methods and code sections are commented	____ 20 points
A method exists to perform the display of output	____ 20 points
A method exists to perform the addition	____ 20 points
Code uses try…catch for exception handling	____ 20 points
Code uses a consistent variable naming convention	____ 10 points
Methods are used appropriately	____ 10 points

Total Points: 335 points

Base K Addition

The simple idea is that the program will be able to perform addition, in decimal, binary, octal, hexadecimal (base 16), or any other base that it is given.

In this problem, you will be given two numbers in base K and asked to add them.

Input:

The input to this problem consists of a text file. The first line of the text file contains a single decimal integer that indicates the number of times the program should perform the addition. The next three lines will repeat for the number indicated in the first line.

The first line of these three numbers contains a single decimal integer indicating the base in which the arithmetic is to be done. This integer will be between 2 and 20 inclusive. The second and third lines of text each consists of a single non-negative number in that base.

In keeping with mathematical tradition, numerals 10-19 (if needed) will be indicated by the letters A-J respectively – upper case only. The sum of the numbers represented in the second and third lines will be smaller than the decimal value 2,014,003,021.

Output:

The output consists of four lines of text. The first line describes the base of the numbers. The second line shows the first number. The third line shows the second number. The fourth line shows the result of the addition.

Example:

```
The base of the numbers to add is 10
The first number is 173
The second number is 169
The sum is: 342
```

Requirements:

1. You must create an application with the main class named baseKaddition.
2. Your name and contestant number must appear as a comment at the top of the main source code file.
3. **The program must read the name of the input file from the command line**. If the file does not exist, then the program needs to display an appropriate message and exit. If the file name is not entered on the command line, then the program should display an appropriate message and exit.
4. The program will perform the addition correctly for however many numbers are in the input file.
5. The program must implement methods for output and the addition.
6. The program will display the output like the example above for all of the values in the input file.