CS1032 Programming Fundamentals (2015 Batch: S1)
Programming Assignment 2 (16 -20 May, 2016)
Q28 (B2)
Time: 1 hour only

Instructions:

1. Save your source file as "p2<StudentNum>.py" in a directory named "p2" in your home directory.
2. Upload the source file, "p2<StudentNum>.py onto the relevant area on Moodle.
3. Write your name and other details in the above box and return this sheet to staff at the end.

Example: If your student number is **"150001C"** then your source file will be **"p2-150001C.py"**

The Problem

Rugby is a popular team sport in the world. You can score in Rugby by the following methods.

- Try – 5 points
- Conversion – 2 points
- Penalty/Drop goal – 3 points

However a team can attempt a conversion only after scoring a try. The team that scored 5 points with a try attempts to add 2 further points. That means a team cannot score 2 points above at any moment, they can either score 5 points (non-converted try) or 5+2=7 points (converted try).

Develop a python program that reads as input the final score (0 < final score < 100) of a team from a file and output the combination of scoring methods that could make up that final score. Output should have only the scoring methods being used and how many of each. If the given final score cannot be achieved in a Rugby Match, output "INVALID SCORE". If it is possible to obtain the final score using multiple combinations, output only the combinations containing the maximum number of converted tries. (Score – 7).

You must also satisfy the following requirements:

(a). Create a text file named "**FileIn.txt**" containing input in the same folder.
(b). Have a Python function named "**getText**" to read input from the "**FileIn.txt**". Invalid inputs handled appropriately.
(c). Have a Python Function named "**showResult**" to display the output on the screen and to write it to a file named "**result.txt**" in the same folder. You may define additional functions in your code.
(d). The main program should call the functions appropriately to achieve the task.

Sample Input 1(Input file):
7
Output (on screen and output file):
```
Try: 1
Conversions: 1
```

Sample Input 2(Input file):
2
Output (on screen and output file):
```
INVALID SCORE
```

Sample Input 1(Input file):
13
Output (on screen and output file):
```
Try: 1
Conversions: 1
Penalty/Drop Goals: 2
```

In this case, another possible combination is "Try: 2, Conversion: 0, Penalty/Drop Goal: 1". But we select the above that has the maximum number of converted tries.

[Fill the top-right box on this page and return this sheet at the end to the staff. Failure to do so is considered as non-submission of the assignment. Do not write anything else unless asked by staff.]

CS1032 Programming Fundamentals (2015 Batch: S1)
Programming Assignment 2 (16 -20 May, 2016)
Q27(B2)
Time: 1 hour only
Instructions:

4. Save your source file as "p2<StudentNum>.py" in a directory named "p2" in your home directory.
5. Upload the source file, "p2<StudentNum>.py onto the relevant area on Moodle.
6. Write your name and other details in the above box and return this sheet to staff at the end.

Example: If your student number is **"150001C"** then your source file will be **"p2-150001C.py"**

<u>The Problem</u>

Some English words have their letters in alphabetical order.
Examples: "almost","effort","first"
We call these words as "In order words".

You are given as input a text file with a set of words on a single line. The length of a word is <20 and the number of words is <1000. Each word may have either lowercase or uppercase letters. Develop a Python program that reads the input file and outputs only the words that are "in order", you can ignore any punctuation marks, digits or non-alphabetical characters in the input.

You must also satisfy the following requirements:
(a). Create a text file named "**FileIn.txt**" containing input in the same folder.
(b). Have a Python function named "**getText**" to read input from the "**FileIn.txt**". Invalid inputs handled appropriately.

(c). Have a Python Function named "**showResult**" to display the output on the screen and to write it to a file named "**result.txt**" in the same folder. You may define additional functions in your code.
(d). The main program should call the functions appropriately to achieve the task.

Sample Input 1 (input file):
```
I am almost in order
```
Output (on screen and output file):
```
I am almost in
```

Sample Input 2 (input file):
```
Coding is fun if you put enough effort
```
Output (on screen and output file):
```
is effort
```

[Fill the top-right box on this page and return this sheet at the end to the staff. Failure to do so is considered as non-submission of the assignment. Do not write anything else unless asked by staff.]

CS1032 Programming Fundamentals (2015 Batch: S1)
Programming Assignment 2 (16 -20 May, 2016)
Q7 (C1)
Time: 1 hour only
Instructions:

1. Save your source file as "p2<StudentNum>.py" in a directory named "p2" in your home directory.
2. Upload the source file, "p2<StudentNum>.py onto the relevant area on Moodle.
3. Write your name and other details in the above box and return this sheet to staff at the end.

Example: If your student number is **"150001C"** then your source file will be **"p2-150001C.py"**

The Problem

Jinna wants to pass a very large positive decimal integer n to Upul. Before doing so Jinna obtains the number p by appending an integer k, which is the number of digits in n, to the right end of n. For example, if n=24687, then k=5 and p=246875. It is this new number p that Jinna will send to Upul.

Accidentally, all the digits in p were shuffled in arbitrary order while it was passed to Upul and Upul receives a number q which is the shuffled version of p. Your task is to help Upul determine n from q. There may be more than one n that can be derived from q, and you should find the smallest possible n. Note that decimal representation of n has no zero digits.

Develop a Python program that reads q from a file as input and outputs the smallest possible integer n which Jinna wanted to pass to Upul. Note that the number of digits in q is <1,000,000. It is guaranteed that the input data is correct, and the answer always exists.

You must also satisfy the following requirements:
(a). Create a text file named "**FileIn.txt**" containing input in the same folder.

(b). Have a Python function named "**getText**" to read input from the "**FileIn.txt**". Invalid inputs handled appropriately.
(c). Have a Python Function named "**show**" to display the output on the screen and to write it to a file named "**result.txt**" in the same folder. You may define additional functions in your code.
(d). The main program should call the functions appropriately to achieve the task.

Sample Input 1 (input file):

Output (on screen and output file):

Sample Input 2 (input file):

Output (on screen and output file):

[Fill the top-right box on this page and return this sheet at the end to the staff. Failure to do so is considered as non-submission of the assignment. Do not write anything else unless asked by staff.]

CS1032 Programming Fundamentals (2015 Batch: S1)
Programming Assignment 2 (16 -20 May, 2016)
Q15 (C3)
Time: 1 hour only
Instructions:

4. Save your source file as "p2<StudentNum>.py" in a directory named "p2" in your home directory.
5. Upload the source file, "p2<StudentNum>.py onto the relevant area on Moodle.
6. Write your name and other details in the above box and return this sheet to staff at the end.

Example: If your student number is **"150001C"** then your source file will be **"p2-150001C.py"**

<u>The Problem</u>

Given a sequence of *k* elements, we obtain its difference sequence by taking the difference between each pair of adjacent elements. That is, the difference sequence of sequence $(a_1, a_2, a_3 \ldots a_k)$ is $(b_1, b_2\ b_3 \ldots b_{k-1})$, where $b_i = a_{i+1} - a_i$.

The derivative sequence of order N o a sequence is the result of repeatedly applying the above process N times. For example, if *A = (5, 6, 3, 9, -1)*, the derivative sequence of order 2 of *A: (5,6,3,9,-1) --> (1, -3, 6, 10) ---> (-4, 9, -16).*

Develop a python program to read as input a sequence A of integers followed by another integer N from a file and output the derivative sequence of order N of the input sequence A. he first line of the input is the sequence A of k integers separated by space and the second line is N. Note that the length of A is k, $1<=k<=100$ and $N<k$.

You must also satisfy the following requirements:
(a). Create a text file named "**FileIn.txt**" containing input in the same folder.
(b). Have a Python function named "**getText**" to read input from the "**FileIn.txt**". Invalid inputs handled appropriately.
(c). Have a Python Function named "**show**" to display the output on the screen and to write it to a file named "**result.txt**" in the same folder. You may define additional functions in your code.
(d). The main program should call the functions appropriately to achieve the task.

Sample Input 1 (input file):
```
4 8 5 6 7 9
1
```
Output (on screen and output file):
```
4 -3 1 1 2
```

Sample Input 2 (input file):
```
5 6 3 9 -1
4
```
Output (on screen and output file):
```
-38
```

[Fill the top-right box on this page and return this sheet at the end to the staff. Failure to do so is considered as non-submission of the assignment. Do not write anything else unless asked by staff.]

CS1032 Programming Fundamentals (2015 Batch: S1)
Programming Assignment 2 (16 -20 May, 2016)
Q16 (C3)
Time: 1 hour only
Instructions:

1. Save your source file as "p2<StudentNum>.py" in a directory named "p2" in your home directory.
2. Upload the source file, "p2<StudentNum>.py onto the relevant area on Moodle.
3. Write your name and other details in the above box and return this sheet to staff at the end.

Example: If your student number is **"150001C"** then your source file will be **"p2-150001C.py"**

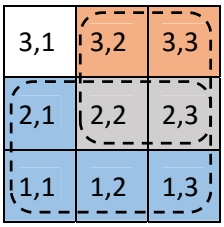
The Problem

Consider a 2-dimensional grid of cells arranged like table, in horizontal rows and vertical columns. The rows are numbered 1, 2, 3 ... bottom-to-top and columns are numbered 1, 2, 3 … right-to-left. Each cell is identified by its (column, row) coordinates, similar to the (x, y) coordinates of a point in X-Y plane. A group of neighboring cells can form a rectangular area on the grid. Such a rectangle can be identified by the coordinates of the lower left cell and upper right cell. For example, in the figure, one rectangle is identified by cells (1, 1) and (2, 3).

You are given a set of rectangles on the grid. You have to compute the number of cells that belong to the rectangles. Note that some rectangles may overlap each other, and as a result, a cell may belong to multiple rectangles, but you should count any cell only once.

For example, in the figure, there are two rectangles:

- rectangle 1: area defined by cells (1, 1) and (2, 3)
- rectangle 2: area defined by cells (2, 2) and (3, 3)



The number of cells that belong to rectangles is 8. The cells (2, 2) (2, 3) belong to both rectangles, but each counted only once.

Develop a python program to read as input a set of rectangles from a file and output the total number of cells that belong to the rectangles, counted as described above. In the input, one rectangle is specified on one line, with 4 integer's separated by space; the first two integers specify the coordinates of the lower-left cell and the next two specify the coordinates of the upper-right cell. Note that the grid size is 100x100.

You must also satisfy the following requirements:

(a). Create a text file named "**FileIn.txt**" containing input in the same folder.

(b). Have a Python function named "**getText**" to read input from the "**FileIn.txt**". Invalid inputs handled appropriately.

(c). Have a Python Function named "**show**" to display the output on the screen and to write it to a file named "**result.txt**" in the same folder. You may define additional functions in your code.

(d). The main program should call the functions appropriately to achieve the task.

Sample Input 1 (input file):

```
1 1 2 3
2 2 3 3
```

Output (on screen and output file):

```
8
```

Sample Input 2 (input file):

```
1 1 3 3
1 1 3 3
```

Output (on screen and output file):

```
9
```