High School Programming Contest

May 12, 2007

Each problem in this packet contains a brief description, followed by two example executions of a successful implementation.

The example input and output shown for each question should be regarded only as a suggestion. We will test your programs on the examples provided, as well as two other sample input test cases.

If your program fails, the result returned by the submission system will state the counterexample test case that caused your program to be judged incorrect. Each incorrect answer will be assessed a 20 minute penalty. As noted in the rules, the overall time will be considered as a tie breaker (with total number of problems solved serving as the initial ranking metric).

Please pay close attention to the directions in each problem description. In some cases, assumptions are stated about the limitations of the input, which are designed so that you do not have to consider difficult cases.

Problem 1:
ROT13

According to Wikipedia, ROT13 ("rotate by 13 places") is a simple Caesar cipher used for obscuring text by replacing each letter with the letter thirteen places down the alphabet.

A becomes N, B becomes O, and so on up to M, which becomes Z, then the sequence reverses: N becomes A, O becomes B, and so on to Z, which becomes M.

The algorithm is used in online forums as a means of hiding joke punchlines, puzzle solutions, movie and story spoilers, and offensive materials from the casual glance.

A noticeable feature of this cipher is that it is symmetrical; that is, to undo ROT13, the same algorithm is applied, so the same code can be used for encoding and decoding.

Your program must be able to apply the ROT13 cipher to its input, and output the result. Only upper case letters A through Z need to be considered, and spaces must be left untouched.

Example 1:

```
Input string:  EXAMPLE ONE
Output string: RKNZCYR BAR
```

Example 2:

```
Input string:  ENIVAR
Output string: RAVINE
```

Problem 2:
Number Speller

Create a program that is able to spell out numbers from 1 to 9999 that are entered in numeric form.

Example 1:

```
Input number:  40
Output string: forty
```

Example 2:

```
Input number:  2007
Output string: two thousand seven
```

Problem 3:
3 by 3 Word Search

This problem considers a list consisting of three lines of three words, and a single search word to lookup within the list. The objective is to output the location of the first occurrence of the single search word within the three lines. This location must include the line number, word number, and character number (the single word may appear as part of another word as in Example 1).

If there are multiple occurrences of the search word, the program should only print the first occurrence.

You may assume that the search word occurs at least once in the list of words.

The lookup must be case sensitive. For instance, “lookup,” “LOOKUP,” and “Lookup” are considered different words.

Example 1:

```
Enter three lines of three words:
Happy Fortieth Anniversary
to UAB CIS
two thousand seven

Enter a single work to lookup:
sand

“sand” found at line 3, word 2, character 5
```

Example 2:

```
Enter three lines of three words:
pink blue red
purple yellow green
red yellow blue

Enter a single work to lookup:
red

“red” found at line 1, word 3, character 1
```

Problem 4:
Roman Numerals Converter

Roman numerals use the following symbols:

I (one)
V (five)
X (ten)
L (fifty)
C (one hundred)
D (five hundred)
M (one thousand)

These are usually written in linear order (e.g., I, II, III) with smaller numerals always following larger numerals except when they are to be subtracted (e.g., IV means 5 - 1). Subtractions are used instead of writing four consecutive occurrences of the same symbol (i.e., IV should be used instead of IIII, MCM instead of MDCCCC, etc.). Only symbols which are powers of 10 may be used in subtractions (i.e., I, X, C, and M). Furthermore, subtraction rules require that a symbol representing 10^x may not precede any symbol larger than 10^(x+1) (e.g., IC is not permitted to represent 99 but XC is 90).

Write a program to input a Roman numeral and output the decimal equivalent if the Roman numeral is well formed and "syntax error" otherwise. Assume that no number will be larger than MMMCMXCIX (3,999).

Example 1:

```
Input: MMVII
Output: 2007
```

Example 2:

```
Input: MIM
Output: syntax error
```

Problem 5:
Genetic Sequencing

DNA is a two-stranded molecule. Each strand is a polynucleotide composed of A (adenosine), T (thymidine), C (cytidine), and G (guanosine) residues. The two strands of DNA run antiparallel (i.e., at each nucleotide residue along the double-stranded DNA molecule, the nucleotides are complementary). That is, A forms two hydrogen-bonds with T; C forms three hydrogen bonds with G. In most cases the two-stranded, antiparallel, complementary DNA molecule folds to form a helical structure, which resembles a spiral staircase. This is the reason why DNA has been referred to as the "Double Helix." An example of two complementary strands of DNA would be:

```
ATGGAATTCTCGCTC
TACCTTAAGAGCGAG
```

This genetic sequencing problem is to find complementary matching strands of DNA. Write a program that reads two strings, and outputs: 1) the beginning part of the second strand which does not match, 2) the part of the second strand which the first string matches, enclosed in -'s, and 3) the remaining part of the second strand which does not match. Note that if there is no match, the second strand will be output with no –'s. If there are multiple matching strings, outputting only the first match is sufficient.

Example 1:

```
Enter 1st strand: ATGC
Enter 2nd strand: ATGCATACGA
Answer: ATGCA-TACG-A
```

Example 2:

```
Enter 1st strand: GAATTC
Enter 2nd strand: TGGAATTCT
Answer: TGGAATTCT
```

Problem 6:
4 x 4 Sudoku

This problem considers a Sudoku puzzle that is to be solved on a 4 x 4 grid, which is subdivided as a 2 x 2 grid of 2 x 2 boxes. The object of the puzzle is to fill in the grid so that every row, every column, and every 2 x 2 box contains the numbers 1 through 4 (i.e., no multiple occurrence of any number is allowed in any row, column, or box). Usually some elements of the grid are already filled in when the puzzle begins.

Write a program to reads 4x4 Sudoku puzzles, with the number 0 representing an unsolved space. The output result should replace all of the 0 spaces in the input puzzle with the final correct solution.

Example 1:

```
Input initial Sudoku puzzle:
0 2 3 0
4 0 2 1
2 1 4 3
3 0 1 2

The solution is:
1 2 3 4
4 3 2 1
2 1 4 3
3 4 1 2
```

Example 2:

```
Input initial Sudoku puzzle:
3 0 4 0
0 4 0 3
2 3 1 4
4 0 3 2

The solution is:
3 2 4 1
1 4 2 3
2 3 1 4
4 1 3 2
```

This page intentionally left blank for scratch paper.

This page intentionally left blank for scratch paper.

This page intentionally left blank for scratch paper.