Sutherland Programming Club

DECEMBER 8, 2017

SAMPLE PROGRAMMING PROBLEM FOR ACSL#1

Problem description

- The main purpose of the problem is to convert input letters to other letters.
- As we can see, 26 letters are divided into 5 intervals, and each interval has its own transform rule.
- Be careful that answer is based on the previous result. We need to add new value to a sum, which can then be converted to an upper case letter.
- The best way might be to attack each interval first, then combine them together using “if” or “case” statement.

Interval #1: A -- E

We can get the final numerical value by simply multiplying original value by 2.

Interval #2: F -- J

First find the remainder of the numerical value after being divided by 3, and then multiply the remainder by 5.

Interval #3: K -- O

First find the integer part after dividing by 4, and then multiply it by 8.

Integer #4: P -- T

Divide the integer by 10 until it becomes 0, and add all the modular value together will be the sum of the digits.

Ex.

	1	2	3	4	5
Integer part	3546	354	35	3	0
Modular part	0	6	4	5	3

The sum of digits will be: 0 + 6 + 4 + 5 + 3 = 18

Interval #5: U -- Z

Find the largest factor by loop from n – 1 to 1; if the a divides n, then a is a factor of n. Return the largest factor * 12.

Ps: visit brute force section if you want to look at how to test whether a number is a prime.

http://sutherland-programming-club.co.nf/previous%20years/16-17/week2.html

Main function

Finally, only need to set a result variable to store the current result. After adding a new return value, we only need to modular result by 26 to keep result in the English letter range.