Practice Arena

Practice problems aimed to improve your coding skills.

- PRACTICE-02_SCAN-PRINT
- PRACTICE-03_TYPES
- LAB-PRAC-02_SCAN-PRINT
- LAB-PRAC-01
- PRACTICE-04_COND
- BONUS-PRAC-02
- LAB-PRAC-03_TYPES
- PRACTICE-05_COND-LOOPS
- LAB-PRAC-04_COND
- LAB-PRAC-05_CONDLOOPS
- PRACTICE-07_LOOPS-ARR
- LAB-PRAC-06_LOOPS
    - Fill in the Square
    - Pretty Numbers
    - Block Cipher
    - The Fibonacci Facade
    - Stream AM GM
    - Int on Int
    - Bejewelled Brooch
    - Mobile Mixup
    - Primes are in C
    - Towering Numbers
    - A Run of One
    - Where are the primes-
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
- LAB-PRAC-08_ARR-STR
- PRACTICE-10_MAT-FUN
- LAB-PRAC-09_PTR-MAT
- LAB-PRAC-10_MAT-FUN
- PRACTICE-11_FUN-PTR
- LAB-PRAC-11_FUN-PTR
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC

Block Cipher

LAB-PRAC-06_LOOPS

Block Cipher [20 marks]

Problem Statement
You will be given a **long integer** N and another **integer** k as input. You have to take the number N and divide it into blocks of length k. For example if the number is 1999567890 and k = 3, then the blocks would be
(1)(999)(567)(890). Note that the last block has only one digit in it.
Given this block structure, we derive an encrypted number as follows: take each block and add the last digit of that block to that block. Take the last k digits of the new number we get as the sum to get the encrypted block. For example, in the above case, we have k = 3, so
890 + 0 = (890)
567 + 7 = (574)
999 + 9 = 1(008)
1 + 1 = (2)
The encrypted number is formed by concatenating all the encrypted blocks i.e.
2008574890

On the first line, print the number of blocks in the integer. In the above example, there are 4 blocks in the number. On the second line print the encrypted number.
Caution

1. Be careful about extra/missing lines and extra/missing spaces.
2. Please note the unusual grading scheme for this question. The grading scheme is described below.

HINTS: Please include the math.h library in your code in case you feel it helps you solve this problem. The log10() function available through the math.h library may help you calculate the number of digits in a number.

EXAMPLE:
INPUT
1999567890 3
OUTPUT:
4
2008574890

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are two lines in your output. Printing each line correctly, in the correct order, carries some weightage. The first line carries 25% weightage and the second line carries 75% weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

Start Solving! (/editor/practice/6112)