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
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
- LAB-PRAC-08_ARR-STR
  - Il fratello di Fibonacci
  - Hidden Palindrome
  - Hush Hush Hash
  - Maximum Match
  - El secreto de sus I
  - Star Replacement
  - Stronger together
  - Rigorous and repeated redaction
  - strnrev
  - Monster and Mini Multiply
  - Clash of the Substrings
  - Personalizing Emails
- 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

Il fratello di Fibonacci

LAB-PRAC-08_ARR-STR

Il fratello di Fibonacci [10 marks]

Problem Statement
We have seen the Fibonacci sequence, and its generalization, the the Lucas sequence, in previous weeks. Today, let us look at another generalization of the Fibonancci sequence where, instead of the first two non-negative integers deciding the sequence, the first k non-negative integers decide the sequence. Let us call this the k-FF-sequence (short for kth order fratello di Fibonacci).

The n-th number in the k-FF sequence is defined to be the sum of the previous k numbers in the k-FF sequence. The first k numbers in the k-FF sequence are 0, 1, ..., k-1. We will give you two strictly positive integers k and n, separated by a single space, as input. As output, you have to output the (k+1)-th number, as well as the n-th number in the k-FF sequence, on the same line but separated by a space.

We promise that k and n will both be less than or equal to 49 and your outputs will always fit inside an int variable.

P.S.: the title of this problem means "Fibonacci's brother" in French.

Caution: Be careful about extra/missing lines and extra/missing spaces in your output.

HINT: Use an array to store the first k k-FF numbers and also calculate the subsequent ones. Remember, we will never give k,n > 100.

EXAMPLE 1:
INPUT
2 5

OUTPUT:
1 3

Explanation: the first two non-negative integers are 0 and 1. So the 2-FF-sequence will be as follows
2-FF1: 0
2-FF2: 1
2-FF3: 0 + 1 = 1
2-FF4: 1 + 1 = 2
2-FF5: 1 + 2 = 3
so the (k+1)-th 2-FF number is the 3rd 2-FF number which is 1 and the 5th 2-FF number is 3.

EXAMPLE 2:
INPUT
3 6

OUTPUT:
3 11

Explanation: the first three non-negative integers are 0, 1, and 2. So the 3-FF-sequence will be as

follows
3-FF1: 0
3-FF2: 1
3-FF3: 2
3-FF4: 0 + 1 + 2 = 3
3-FF5: 1 + 2 + 3 = 6
3-FF6: 2 + 3 + 6 = 11
so the (k+1)-th FF number is the 4th 3-FF number which is 3 and the 6th 3-FF number is 11.

Grading Scheme:
Total marks: **[10 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

Start Solving! (/editor/practice/6163)