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
  - The D List
  - The D Factor
  - All Charged Up
  - The S Factor
  - The S List
  - Smith Numbers
- 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

The S List

LABEXAM-PRAC-01_MIDSEM

The S List [50 marks]

Problem Statement
We will give you a **strictly positive integer** n. Given this, you have to print your output **in two separate lines**

1. In the first line, print a list of all prime numbers strictly less than the square root of n. Two primes should be separated by a space. **There should be no trailing spaces at the end of this line**. If there are no such primes, print "NO PRIMES" (without quotes).
2. In the second line print the sum of the values of nCk for all k that are primes strictly less than the square root of n. If there are no such primes, print 0.

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Problem-specific Words of Caution:

1. Take care that combinatorial expressions involve factorials that can take huge values that can even exceed the long limit.
2. Even though your input will fit inside an integer variable, your output may require long variables.

EXAMPLE:
INPUT
7

OUTPUT:
2
21

Explanation: sqrt(7) = 2.64. Thus the only prime less than sqrt(7) is 2 and 7 choose 2 is 21.

General Words of Caution

1. **Do not forget to submit your code**. You can submit multiple times. Your last submission will get graded.
2. Marks will be allotted for the following

    1. Proper and meaningful variable names
    2. Nice looking and consistent indentation
    3. At least a couple of comments explaining to the human grader what are you doing, especially when the calculations are not obvious
    4. Comments, good indentation and meaningful variable names are very important for the human grader to understand what are you doing and why. If they cannot understand your code, do not expect them to give you (partial) marks either.

3. Solutions that indulge in hard-coding **will get a straight zero** even if they are passing some test cases. Hard-coding is a form of cheating strategy where someone write code of the form "if(input == A) printf(B)" without doing any calculations on A to obtain B. The values of A and B are either read from the evaluation/submission window or else guessed.
4. Questions will be graded by the **autograder as well as a human grader**
5. Be careful about extra/missing lines and extra/missing spaces if you do not want to lose autograder marks
6. Proportion of marks allotted to autograder (in particular, weightage to visible and hidden test cases) and human grader will be revealed when marks and grading rubrics are released
7. The total marks of this exam are 150.
8. You are allowed to use the libraries math.h and stdlib.h **but not any other library**. Use of unpermitted libraries will carry a penalty. You may use programming tools such as arrays, functions, recursion, pointers, in case you are familiar with the use of these. However, you will be given no special credit for using these advanced programming techniques, nor will you receive any help should you face difficulties in using them, for example, TLE or segmentation fault errors. Use these advanced techniques at your own risk.

Grading Scheme:
Total marks: **[50 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 has 25% weightage and the second line has 75% weightage. There are 3 visible and 6 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/6156)