


































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_CONDLLOOPS
-  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

Smith Numbers

LABEXAM-PRAC-01_MIDSEM

Smith Numbers [75 marks]

Problem Statement

Smith numbers are composite numbers for which the sum of the digits of the number is equal to the sum of the digits of all its prime factors (including repetitions).

For example, $378 = 2 \times 3 \times 3 \times 3 \times 7$ is a Smith number since $3 + 7 + 8 = 2 + 3 + 3 + 3 + 7$. Another nice example of a Smith number is $22 = 2 \times 11$ and is a Smith number since $2 + 2 = 2 + 1 + 1$.

You will be given a **strictly positive integer** n . You have to print "Smith Number" (without quotes) in case the number is a Smith number else print "Not Smith Number" (without quotes) in case the number is not a Smith number.

Problem-specific Words of Caution:

1. Take care of spelling and other mistakes
 2. As mentioned in the problem statement above, only composite numbers can be Smith numbers, i.e. prime numbers cannot be Smith numbers.
-

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: **[75 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). There are 4 visible and 8 hidden test cases.

 **Start Solving! (/editor/practice/6157)**