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 D List

LABEXAM-PRAC-01_MIDSEM

The D List [25 marks]

Problem Statement
We will give you a list of digits given as **non-negative integers** all separated by a space. The list will be ended by a -1 which is not supposed to be considered a part of the list. You have to process this list and output the following **on different lines**

1. The number of elements in the list (not including the last -1)
2. The number of times the digit 2 appears in the list
3. The number of times the first digit of the list appears in the list
4. If the number formed by the list is divisible by 11, print "YES" (without quotes), else print "NO" without quotes

HINTS: To test the divisibility of a number by 11, take an alternating sum of the digits of the number. For example, if the number is 1234, an alternating sum would be 1 - 2 + 3 - 4. If this sum is divisible by 11, so is the original number. If this sum is not divisible by 11, neither is the original number.

EXAMPLE:
INPUT
3 2 -1

OUTPUT:
2
1
1
NO

Explanation: The list has two digits, the digit 2 appears once, the first digit 3 also appears just once, and the number 32 is not divisible by 11.

Problem-specific Words of Caution:

1. The numbers 0, -22, 110, -44 etc are all considered divisible by 11. However, then numbers -4, 10, 16 etc are not divisible by 11. In particular, note that negative numbers can be divisible by 11 as well.
2. We promise that the first digit will never be zero, as well as that the number formed by the digits will be strictly positive
3. The number formed by the digits we give you may not fit inside int or long variables.

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: **[25 Points]**

There will be partial grading in this question. There are four lines in your output. Printing each line correctly, in the correct order, carries some weightage. The first two lines are worth 12.5% each. The third line is worth 25% and the fourth line is worth 50% weightage. 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/6152)