Common Instructions

There are 3 components of the marking scheme:

1. **Autograde Score**: given by autograder based on performance on test cases
    a. out of 10 for 25 marks questions (1 + 1 + 2 + 2 + 2 + 2 for 6 test cases)
    b. out of 21 for 50 marks questions (1 + 1 + 1 + 3 + 3 + 3 + 3 + 3 + 3 for 9 test cases)
    c. out of 36 for 75 marks questions (1 + 1 + 1 + 1 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 for 12 test cases)
2. **Manual Score**: given by human graders
    a. out of 10 for 25 marks questions (see questions for break up)
    b. out of 20 for 50 marks questions (see questions for break up)
    c. out of 30 for 75 marks questions (see questions for break up)
3. **Good Coding Score**: given by human graders for indentation, proper variable names, comments
    a. out of 5 for 25 marks questions (2 indentation + 2 variable names + 1 comments)
    b. out of 9 for 50 marks questions (4 indentation + 3 variable names + 2 comments)
    c. out of 9 for 75 marks questions (4 indentation + 3 variable names + 2 comments)
4. **Penalty Score:** given for illegal library use and hardcoding.
    a. Solutions that indulge in **hard-coding get a straight zero overall** even if passing some test cases. Hard-coding is a form of cheating where someone writes 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 read from evaluation/submission window or else guessed. Solutions that only do hardcoding for edge cases (if(n == 2) printf("PRIME");) are still given penalties.
    b. Only libraries permitted are stdio.h, math.h and stdlib.h. If any other libraries are used (e.g. string.h), a penalty is given
    c. Use of arrays, functions, recursion, pointers is allowed. No penalties for using these.

Note: only submissions will full Autograder score can get full manual score. A few marks in manual score are reserved for those who got full autograder score. Marks reserved are less if there was less scope for partial grading e.g. in the Smith Number's question where autograding was harsh since there was no partial grading, or if question was harder.

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

Rubric

1. **Penalty Score**: hardcoding penalty score is **-25**, illegal library use penalty score is **-5**.
2. **Good Coding Score**: **2 marks** for good indentation, **2 marks** for proper meaningful variable names and **1 mark** for informative comments.

3. **Manual Score**: (only submissions will full Autograder score can get full manual score. 3 marks in manual score are reserved for those who got full autograder score.)
    a. If there is no penalty and Autograder Score is 10, **10 marks** in Manual Score, else
    b. Give **2 marks** for defining and properly using variables to keep track of running count of how many 2s were seen
    c. Give **2 marks** for defining and properly using variables to keep track of running count of how many times the first digit was seen
    d. Give **2 marks** for defining and properly using a variable to keep track of the alternating sum
    e. Give **1 marks** for implementing a check for when to stop reading numbers
    f. In the above, if the steps are carried out, but with major flaws or outright wrong logic, give partial marks.

The D Factor (50 marks)

Problem Statement You will be given a strictly positive integer n. In two separate lines print the following

1. On the first line print the distinct divisors of n in increasing order with two divisors separated by a space. Divisors include prime and non-prime divisors as well as 1 and the number itself. There should be no trailing spaces at the end of this line
2. On the second line, print the smallest strictly positive number M that has exactly n distinct divisors (including prime and non-prime divisors and including 1 and the number M itself)

Rubric

1. **Penalty Score**: hardcoding penalty score is **-50**, illegal library use penalty score is **-10**.
2. If Penalty score is being awarded, please mention reason in Penalty Reason. If no Penalty Score then leave Penalty Score blank and Penalty Reason blank.
3. **Good Coding Score**: give **4 marks** for good indentation, **3 marks** for proper meaningful variable names and **2 marks** for informative comments.
4. **Manual Score**: (only submissions will full Autograder score can get full manual score. 5 marks in manual score are reserved for those who got full autograder score.)
    a. If there is no penalty and Autograder Score is 21, give **20 marks** in Manual Score, else
    b. Give **3 marks** for defining variables and properly using a loop to detect divisors of n and printing all of them.
    c. Give **3 marks** for maintaining a variable to keep track of what is the number of divisors of the current number.
    d. Give **4 marks** for using an outer loop to check for various numbers for their distinct factors
    e. Give **5 marks** for using an inner (nested loop) to check within an outer loop, what is the number of distinct factors of the number being currently investigated
    f. In the above, if the steps are carried out, but with major flaws or outright wrong logic, give partial marks.

All Charged Up (75 marks)

Problem Statement Electrostatic charges (either +1 or -1) have been placed at integer coordinates in a 3D space. The x, y, z, coordinates of these charges can all range from 1 to 256 (both limits included). The charge at a given location (i,j,k) is determined by the following rules.

1. Case 1: if the location is at the boundary of this arrangement (i.e. outer surface of this 256 x 256 x 256 sized 3D cube), the charge at that location is +1
2. Case 2: if the location is not at the boundary but the sum i + j + k is prime, then also the charge at that location is +1
3. Case 3: if the location is neither at the boundary, nor is the sum of coordinates prime, but still the coordinates (i,j,k) satisfy the property that the quadratic equation i * x^2 + j * x + k has only real roots, even then the location has charge +1
4. Case 4: any location that does not have charge +1 has charge -1

We will give you two locations (a,b,c) and (p,q,r) with all coordinates in the range [1, 256] (both ends included). Consider the smallest cuboid C that contains both these locations. Your task is to print the following in five separate lines.

1. The number of locations inside or on the boundary of the cuboid C that satisfy case 1
2. The number of locations inside or on the boundary of the cuboid C that satisfy case 2
3. The number of locations inside or on the boundary of the cuboid C that satisfy case 3
4. The number of locations inside or on the boundary of the cuboid C that satisfy case 4
5. The sum of all charges at locations inside or on the boundary of the cuboid C

Rubric

1. **Penalty Score**: hardcoding penalty score is **-75**, illegal library use penalty score is **-15**.
2. If Penalty score is being awarded, please mention reason in Penalty Reason. If no Penalty Score then leave Penalty Score blank and Penalty Reason blank.
3. **Good Coding Score**: give **4 marks** for good indentation, **3 marks** for proper meaningful variable names and **2 marks** for informative comments.
4. **Manual Score**: (only submissions will full Autograder score can get full manual score. 9 marks in manual score are reserved for those who got full autograder score.)
    a. If there is no penalty and Autograder Score is 36, give **30 marks** in Manual Score, else
    b. Give **5 marks** for using conditional/relational statements to find out the smallest cuboid that contains the two points.
    c. Give **3 marks** for using a triple nested loop to go over all points in the cuboid.
    d. Give **3 marks** for correct check for boundary points
    e. Give **3 marks** for correct check for points with prime sum
    f. Give **3 marks** for correct check for points with real roots
    g. Give **3 marks** for using an if-else statement to count how many points of each type are thre in the cuboid.
    g. Give **1 mark** for correctly calculating total charge in the cuboid.
    h. In the above, if the steps are carried out, but with major flaws or outright wrong logic, give partial marks.

The S Factor (25 marks)

Problem Statement You will be given a long integer n as input. Using this integer, you have to print the following on four different lines

1. The smallest divisor of n that is strictly greater than 1 (notice that this divisor has to be prime)
2. The number of times the digit 3 appears when we write down n
3. The number of times the first digit of n (when seen from the left) appears when we write down n
4. The power of the smallest divisor of n in its prime factorization

Rubric

1. **Penalty Score**: hardcoding penalty score is **-25**, illegal library use penalty score is **-5**.
2. If Penalty score is being awarded, please mention reason in Penalty Reason. If no Penalty Score then leave Penalty Score blank and Penalty Reason blank.
4. **Good Coding Score**: give **2 marks** for good indentation, **2 marks** for proper meaningful variable names and **1 mark** for informative comments.
5. **Manual Score**: (only submissions will full Autograder score can get full manual score. 1 mark in manual score are reserved for those who got full autograder score.)
    a. If there is no penalty and Autograder Score is 10, give **10 marks** in Manual Score, else
    b. Give **2 marks** for using and properly stopping a loop to find the smallest divisor of n
    c. Give **2 marks** for using a loop to find out the power of that loop in prime factorization of n
    d. Give **1 marks** for defining and properly using a variable and a loop to keep track of the number of times 3 appears when we write down n
    e. Give **2 marks** for finding the first digit of n
    f. Give **2 marks** for using a loop to find number of occurrences of the first digit of n
    g. In the above, if the steps are carried out, but with major flaws or outright wrong logic, give partial marks.

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 $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ for all k that are primes strictly less than the square root of n. If there are no such primes, print 0.

Rubric

1. **Penalty Score**: hardcoding penalty score is **-50**, illegal library use penalty score is **-10**.
2. If Penalty score is being awarded, please mention reason in Penalty Reason. If no Penalty Score then leave Penalty Score blank and Penalty Reason blank.
3. **Good Coding Score**: give **4 marks** for good indentation, **3 marks** for proper meaningful variable names and **2 marks** for informative comments.

4. **Manual Score**: (only submissions will full Autograder score can get full manual score. 2 marks in manual score are reserved for those who got full autograder score.)
    a. If there is no penalty and Autograder Score is 21, give **20 marks** in Manual Score, else
    b. Give **2 marks** for correct code to detect prime numbers
    c. Give **3 marks** for properly using an outer loop to print primes less than sqrt(n)
    d. Give **5 marks** for using scalable code i.e. $\binom{n}{k} = \frac{n \cdot n-1 \cdot \ldots \cdot (n-k+1)}{k!}$
    e. Give only **2 marks** if using bad code that calculates n! and (n-k)! explicitly and divides them
    f. Give **3 marks** for handling the edge case of prime < root vs prime <= root (Q asks for former)
    g. Give **3 marks** for handling the edge case when the answer is NO PRIMES
    h. In the above, if the steps are carried out, but with major flaws or outright wrong logic, give partial marks.

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 × 3 × 3 × 3 × 7 is a Smith number since 3 + 7 + 8 = 2 + 3 + 3 + 3 + 7. Another nice example of a Smith number is 22 = 2 × 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) otherwise.

Rubric

1. **Penalty Score**: hardcoding penalty score is **-75**, illegal library use penalty score is **-15**.
2. If Penalty score is being awarded, please mention reason in Penalty Reason. If no Penalty Score then leave Penalty Score blank and Penalty Reason blank.
3. **Good Coding Score**: give **4 marks** for good indentation, **3 marks** for proper meaningful variable names and **2 marks** for informative comments.
4. **Manual Score**: (only submissions will full Autograder score can get full manual score. 2 marks in manual score are reserved for those who got full autograder score.)
    a. If there is no penalty and Autograder Score is 36, give **30 marks** in Manual Score, else
    b. Give **5 marks** for checking the special case of prime numbers and handling that.
    c. Give **5 marks** for code for detecting prime divisors of n
    d. Give **6 marks** for code for checking what is the power of a certain prime in factorization of n
    e. Give **5 marks** for code to calculate sum of digits of a prime divisor
    f. Give **5 marks** for code to calculate sum of digits of n
    g. Give **2 marks** for correctly calculating sum of digits of all prime divisors **with repetition**
    h. In the above, if the steps are carried out, but with major flaws or outright wrong logic, give partial marks.