

Digit Frequency Count Problem

Problem Statement

Given a string consisting of alphabets and digits, find the frequency of each digit (from 0 to 9) in the string.

Input Format

The first line contains a string S which contains alphabets and digits.

Constraints

- All elements of S are English alphabets or digits.

Output Format

Print ten space-separated integers denoting the frequency of digits from 0 to 9 in the input string.

Sample Input 0

a11472o5t6

Sample Output 0

0 2 1 0 1 1 1 1 0 0

Explanation 0

- Digit 1 occurs two times.
- Digits 2, 4, 5, 6, and 7 occur once each.
- Digits 0, 3, 8, and 9 do not occur.

Solution in C

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <ctype.h>
4
5 int main() {
6     char str[1000];
7     int freq[10] = {0};
8
9     scanf("%s", str);
10
```

```

11     for (int i = 0; i < strlen(str); i++) {
12         if (isdigit(str[i])) {
13             freq[str[i] - '0']++;
14         }
15     }
16
17     for (int i = 0; i < 10; i++) {
18         printf("%d ", freq[i]);
19     }
20
21     return 0;
22 }

```

Listing 1: Digit Frequency Count in C

Explanation

- We read the input string into `str`.
- We create an integer array `freq` of size 10 initialized to 0 to keep track of digit frequencies.
- We iterate over each character in `str`, check if it is a digit using `isdigit()`.
- If yes, convert the character digit to its integer value by subtracting `'0'` and increment the respective count in `freq`.
- Finally, print the frequency of each digit separated by spaces.

Common Similar Problems

- Counting frequency of alphabets in a string.
- Counting frequency of both digits and alphabets in a string.
- Finding the most frequent digit or character.
- Counting frequency of characters ignoring case sensitivity.
- Counting frequency of special characters in addition to digits and alphabets.

How to Make the Problem Harder

- **Multi-line Input:** Instead of a single line, process multiple lines of input strings.
- **Large Inputs:** Handle very large input efficiently (e.g., up to 10^7 characters).
- **Case Sensitivity:** Count uppercase and lowercase alphabets separately.
- **Full Character Set:** Count digits, alphabets, and special characters.
- **Dynamic Input:** Read input until EOF or a termination string is reached.
- **Output Format Variations:** Print results as a histogram or sorted by frequency.
- **Memory Constraints:** Solve the problem with limited memory (streaming input).
- **Non-ASCII Characters:** Handle Unicode or multibyte characters.

- **Interactive Mode:** Allow queries like "How many times does digit X appear?" after the input.