Is Hashing Always Good ?

Hashing gives a more secure and adjustable method of retrieving data than any other data structure. However, it is not suitable if the hashing technique outputs many collisions. Sirimath is an employee at the Apollo cyber security firm. He was tasked to check whether the given hash function was better to use. As a programmer, you should help Sirimath to achieve that. Your job is to output the number of collisions occurring in all permutations of a given string s of length n.

Collision: two pieces of data share the same hash value.

Permutation: A permutation is an ordered arrangement of objects. For example: 3124, 1423 is two possible permutation of the digits 1, 2, 3 and 4.

Hash Function Description

Implement the following hash function efficient manner.

$$\text{hash}(s) = (s[0] + s[1] \cdot p + s[2] \cdot p^2 + \ldots + s[n-1] \cdot p^{n-1}) \mod m$$

$$p = 5, m = 10^n$$

$s[i]$ = Ascii value of ith character of string s.

Note: If n permutations result in the same hash value, we can only use one permutation, the other $n - 1$ permutations will be considered as collisions.

Input Format

Single string s consisting of characters $a - z, A - Z, 0 - 9$.

Constraints

$1 < n < 11$

Output Format

Print the number of collisions in all permutations of the given string.

Sample Input 0

```
asxz
```

Sample Output 0

```
1
```

Explanation 0

Permutations 'axzs' and 'zsax' produce the same hash value "8122". Therefore one collision occurs.

Sample Input 1

```
pb159
```

Sample Output 1

```
1
```

Explanation 1

Permutations 'b5p19' and '5p9b1' produce the same hash value "44913". Therefore one collision occurs.