848. Shifting Letters

DescriptionHintsSubmissionsDiscussSolution

We have a string `S` of lowercase letters, and an integer array `shifts`.

Call the *shift* of a letter, the next letter in the alphabet, (wrapping around so that `'z'` becomes `'a'`).

For example, `shift('a') = 'b'`, `shift('t') = 'u'`, and `shift('z') = 'a'`.

Now for each `shifts[i] = x`, we want to shift the first `i+1` letters of `S`, `x` times.

Return the final string after all such shifts to `S` are applied.

Example 1:

```
Input: S = "abc", shifts = [3,5,9]
Output: "rpl"
Explanation:
We start with "abc".
After shifting the first 1 letters of S by 3, we have "dbc".
After shifting the first 2 letters of S by 5, we have "igc".
After shifting the first 3 letters of S by 9, we have "rpl", the answer.
```

Note:

1. `1 <= S.length = shifts.length <= 20000`
2. `0 <= shifts[i] <= 10 ^ 9`

Seen this question in a real interview before?

- Difficulty:Medium
- Total Accepted:3K
- Total Submissions:8.9K
- Contributor:awice
-
- Subscribe to see which companies asked this question.

  Related Topics String

Straight Forward Approach

```
class Solution {
public:

string shiftingLetters(string S, vector<int>& shifts) {
   int n = shifts.size();
   vector<int> shifts_new(n,0);
   reverse(shifts.begin(),shifts.end());
   int ret = 0;
   for(int i=0;i<n;++i)
   {
      ret+=shifts[i]%26;
      shifts_new[i]=ret;
   }
   reverse(shifts_new.begin(),shifts_new.end());
   //for(int i=0;i<n;i++) printf("%d\n",shifts_new[i]);
   for(int i=0;i<(int)S.size();++i)
   {
      shifts_new[i]=shifts_new[i]%26;
      if(S[i]+shifts_new[i]<='z')
      {
         S[i] = S[i]+shifts_new[i];
      }else{
         S[i] = (S[i]+shifts_new[i]-'z')+'a'-1;
      }
   }
   return S;
}


};
```