

Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8201512	15:17:20	Suffix Sorting	✓ Accepted	0.17 s	C++
	✓✓				

Submission contains 1 file:

download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
suffixsorting.cpp	2398 bytes	e65400efc6d120fdd4d463bfff131319d97fb94d4	download

Edit and resubmit this submission.

suffixsorting.cpp

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 #define INF 999999
6 #define FOR(A,B,C) for(int i=A; i<B; i+=C)
7 #define FOR_bit(A,B,C) for(int a=A; a<B; a<=C)
8
9 vector<int> arrSufijos(INF);
10 vector<int> arrID(INF);
11     nt> ordenActual(INF);
12
13 void llenar(int n, string s){
```

```

14     FOR(0,n,1){
15         arrSufijos[i] = i;
16         arrID[i] = s[i];
17     }
18 }
19
20 void actualizar(int n, int maxBytes){
21     FOR_bit(1,2*n,1) {
22
23         int mitad = a/2;
24
25         FOR(0,n,1){
26             ordenActual[i] = arrSufijos[i] - mitad;
27             if (ordenActual[i] < 0){
28                 ordenActual[i] += n;
29             }
30         }
31
32         vector<int> aux(maxBytes);
33
34         int it;
35         FOR(0,n,1){
36             it = arrID[ordenActual[i]];
37             aux[it]++;
38         }
39
40         FOR(1,maxBytes,1){
41             aux[i] = aux[i]+aux[i - 1];
42         }
43
44         int it2;
45         FOR(0,n,1){
46             it2 = ordenActual[n-i-1];
47             aux[arrID[it2]]--;
48             it = aux[arrID[it2]];
49             arrSufijos[it] = it2;
50         }
51
52         vector<int> nuevoArrID(n);
53         maxBytes = 1;
54
55         Help nuevoArrID[arrSufijos[0]] = 0;
56
57         FOR(1,n,1){

```

```

58         if(arrID[arrSufijos[i]] == arrID[arrSufijos[i - 1]]){
59             int pos1 = arrSufijos[i] + mitad;
60             if (pos1 > n){
61                 pos1 -= n;
62             }
63             int pos2 = arrSufijos[i - 1] + mitad;
64             if (pos2 > n){
65                 pos2 -= n;
66             }
67             if (arrID[pos1] != arrID[pos2]){
68                 maxBytes++;
69             }
70         }
71         else{
72             maxBytes++;
73         }
74
75         nuevoArrID[arrSufijos[i]] = maxBytes - 1;
76     }
77
78     arrID = nuevoArrID;
79 }
80 }
81
82 vector<int> obtenerArregloSufijos(string s) {
83
84     int n = s.size() + 1;
85     int maxBytes = 1024;
86
87     llenar(n, s);
88     actualizar(n, maxBytes);
89
90     FOR(0,n-1,1){
91         arrSufijos[i] = arrSufijos[i + 1];
92     }
93
94     //Debido a que se trabaja con size + 1
95     arrSufijos.pop_back();
96
97
98     return arrSufijos;
99
100
101 int main(){

```

```
102     string cad;
103     int querie;
104     int n; //Casos
105     vector<int> sufijos;
106
107     while(getline(cin, cad)){
108
109         sufijos = obtenerArregloSufijos(cad);
110
111         cin>>n;
112         for(int i=0; i<n; i++){
113
114             cin>>querie;
115             cout<<sufijos[querie]<<" ";
116         }
117         cout<<endl;
118         cin.ignore();
119
120         sufijos.clear();
121     }
122
123
124
125
126     return 0;
127 }
```