



# **Substring Order I**

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

# **Submission details**

Task:	<u>Substring Order I</u>
Sender:	Rodry
Submission time:	2021-12-09 21:46:39
Language:	C++17
Status:	READY
Result:	ACCEPTED

# **Test results** ▲

test	verdict	time	
#1	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#2	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#3	ACCEPTED	0.12 s	<u>&gt;&gt;</u>
#4	ACCEPTED	0.12 s	<u>&gt;&gt;</u>
#5	ACCEPTED	0.06 s	<u>&gt;&gt;</u>
#6	ACCEPTED	0.05 s	<u>&gt;&gt;</u>
#7	ACCEPTED	0.14 s	<u>&gt;&gt;</u>
#8	ACCEPTED	0.05 s	<u>&gt;&gt;</u>
#9	ACCEPTED	0.12 s	<u>&gt;&gt;</u>

# Code ▲

1	<pre>#include<bits stdc++.h=""></bits></pre>		
2	<pre>#include<bits stdc++.h=""> using namespace std;</bits></pre>		
3			
4	#define INF 9999999		

# **String Algorithms**

• • •	
Counting Patterns	_
Pattern Positions	_
Distinct Substrings	_
Repeating Substring	_
String Functions	_
Substring Order I	<b>✓</b>
Substring Order II	×
Substring Distribution	_

## **Your submissions**

2021-12-09 21:46:39	<b>✓</b>
2021-12-09 21:39:54	×
2021-12-09 21:38:05	X

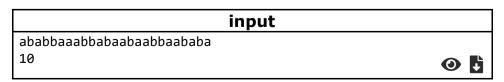
```
5
 6 long long sa[INF], pos[INF], tmp[INF], lcp[INF];
 7 long long gap, N;
 8 string S;
10 | bool comp(long long x, long long y) {
       if (pos[x] != pos[y])
11
12
           return pos[x] < pos[y];</pre>
13
       x += gap;
14
       y += gap;
15
       return (x < N && y < N)? pos[x] < pos[y] : x > y;
16 }
17
18 void sufijos() {
19
       for (int i = 0; i < N; i++)
20
           sa[i] = i, pos[i] = S[i];
21
22
       for (gap = 1;; gap <<= 1) {</pre>
23
           sort(sa, sa+N, comp);
24
           for (int i = 0; i < N-1; i++)
25
               tmp[i+1] = tmp[i] + comp(sa[i], sa[i+1]);
26
           for (int i = 0; i < N; i++)
27
               pos[sa[i]] = tmp[i];
28
           if (tmp[N - 1] == N - 1)
29
               break;
30
31
32
   void construirLCP() {
33
34
       for (int i = 0, k = 0; i < N; i++)
35
                    if (pos[i] != N-1) {
36
           long long j = sa[pos[i] + 1];
37
           while (S[i + k] == S[j + k])
38
               k++;
39
           lcp[pos[i]] = k;
40
           if (k)
41
                            k--;
42
43 }
44
45 int main(){
46
47
48
       cin>>S;
49
           N = S.size();
50
```

```
51
           sufijos();
52
53
       construirLCP();
54
55
       long long k;
56
           cin>>k;
57
58
           long long prev = 0;
       long long act = 0;
59
60
61
62
       for (int i = 0; i < N; i++) {
63
           if (act + (N-sa[i]) - prev >= k) {
64
               long long j = prev;
65
               string ans = S.substr(sa[i], j);
               while (act < k) {</pre>
66
67
                    ans += S[sa[i]+j];
68
                   act++;
69
                   j++;
70
71
               cout<<ans;
72
               return 0;
73
74
           act += (N-sa[i]) - prev;
75
           prev = lcp[i];
76
77 }
```

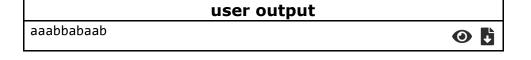
#### Share code to others

# Test details ▲

## Test 1



	correct output	
aaabbabaab		<b>O</b>



Test 2

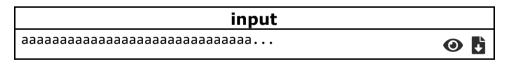
Verdict: ACCEPTED

input	
rmnxvouggsdgespsltsldcvkxtg	
33	<b>O</b>

	correct output
esps	<b>② 5</b>

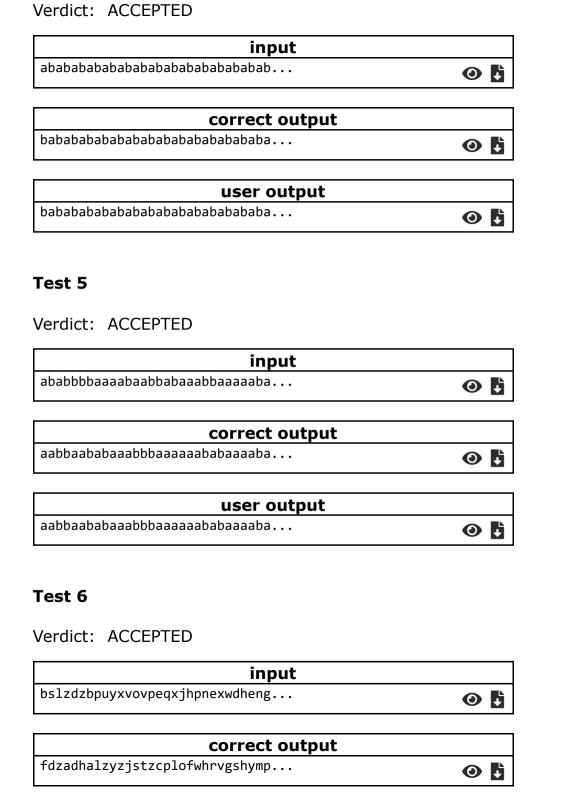
	user output	
esps		<b>0</b>

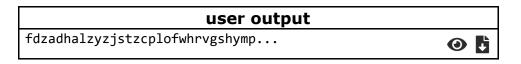
# Test 3



correct output	
aaaaaaaaaaaaaaaaaaaaa	<b>0</b>

user output	
aaaaaaaaaaaaaaaaaaaa	<b>©</b>





## Test 7

Verdict: ACCEPTED

input	
iybvzbbtkfqevdtjwqhezljzdkkjwi	<b>⊙</b>

correct output	
brdcxlfbsneugpmevkwmehndrzncoh	<b>O</b>

user output	
brdcxlfbsneugpmevkwmehndrzncoh	•

# Test 8



correct output	
phtjwgbtgbhslxxtprgbyppsnekyoy	•

user output	
phtjwgbtgbhslxxtprgbyppsnekyoy	<b>©</b>

input	
aaaaaaaaaaaaaaaaaaaaaa	<b>O</b>

correct output	
aaaaaaaaaaaaaaaaaaaaaa	<b>0</b>

user output	
aaaaaaaaaaaaaaaaaaaaaa	<b>0</b>