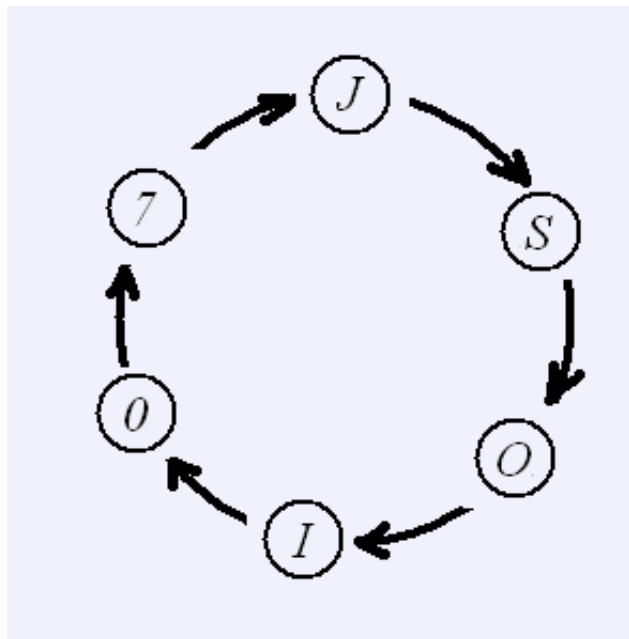


题意

喜欢钻研问题的JS同学，最近又迷上了对加密方法的思考。一天，他突然想出了一种他认为是终极的加密办法

：把需要加密的信息排成一圈，显然，它们有很多种不同的读法。例如下图，可以读作：



JSOI07 SOI07J OI07JS IO7JSO 07JSOI 7JSOI0

把它们按照字符串的大小排序：07JSOI 7JSOI0 IO7JSO JSOI07 OI07JS SOI07J

读出最后一列字符：IO07SJ，就是加密后的字符串。

但是，如果想加密的字符串实在太长，你能写一个程序完成这个任务吗？

对于100%的数据字符串的长度不超过100000。

思路

将字符串s扩成两倍ss

计算出ss的**后缀数组**sa[]

遍历sa[] 数组，如果sa[i] >= s.size() 跳过

否则输出 s[sa[i] + s.size() - 1]

参考资料：刘汝佳蓝书 后缀数组模板

代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long LL;
4  const int MAXN = 200010;
5  int rank[MAXN];
6  int t[MAXN],t2[MAXN],c[MAXN],n;
7  int sa[MAXN];
8  string s;
9  void build_sa(int m)
10 {
11     int i,*x = t,*y = t2;
12     n = s.size();
13     for(i = 0 ;i<m;i++) c[i] = 0;
14     for(i = 0;i<n;i++) c[x[i] = s[i]] ++;
15     for(i = 1;i<m;i++) c[i]+=c[i-1];
16     for(i = n-1;i>=0;i--) sa[--c[x[i]]] = i;
17     for(int k=1;k<=n;k<=1)
18     {
19         int p=0;
20         for(i=n-k;i<n;i++) y[p++] = i;
21         for(i=0;i<n;i++)
22         {
23             if(sa[i]>=k) y[p++] = sa[i]-k;
24         }
25         for(i=0;i<m;i++)c[i] = 0;
26         for(i=0;i<n;i++)c[x[y[i]]]++;
27         for(i=0;i<m;i++) c[i] +=c[i-1];
28         for(i=n-1;i>=0;i--) sa[--c[x[y[i]]]] = y[i];
29         swap(x,y);
30         p=1;
31         x[sa[0]] = 0;
32         for(i=1;i<n;i++)
33         {
34             x[sa[i]] = y[sa[i-1]] == y[sa[i]] && y[sa[i-1]+k] == y[sa[i]+k]?p-
1:p++;
35         }
36         if(p>=n) break;
37         m=p;
38     }
39 }
40 int main()
41 {
```

```
42     cin>>s;
43     s = s+s;
44     build_sa(128);
45
46     for(int i=0;i<n;i++)
47     {
48         if(sa[i] >= (n/2)) continue;
49         int kk = sa[i]-1;
50         if(kk==-1) kk=n-1;
51         cout<<s[kk];
52     }
53 }
```