# Dance

3074

#include<bits/stdc++.h>

using namespace std;

const int M = 300;

char s[M];//原来数字

int s2[M];//后来填上的数字

struct{

int l,r,d,u;

int val;

int x,y;

}P[M\*M];

int pn;

int last[M\*4];

int build(){//324是列数

for(int i=0;i<=324;i++){

P[i+1].l=i;

P[i].r=i+1;

P[i].val=0;

last[i]=i;

//建立列首 P[i].val保存该列有多少个元素

}

pn=324;

P[324].r=0;

P[0].l=324;

return 0;

}

int del(int num){

P[P[num].d].u=P[num].u;

P[P[num].u].d=P[num].d;

return 0;

}

int redel(int num){

P[P[num].d].u=num;

P[P[num].u].d=num;

return 0;

}

int modefy1(int c){

P[P[c].r].l=P[c].l;

P[P[c].l].r=P[c].r;

for(int i=P[c].d;i!= c;i=P[i].d){

for(int j=P[i].r;j!=i;j=P[j].r){

del(j);

P[P[j].y].val--;

}

}

return 0;

}

int modefy2(int c){

P[P[c].r].l=c;

P[P[c].l].r=c;

for(int i=P[c].d;i!= c;i=P[i].d){

for(int j=P[i].r;j!=i;j=P[j].r){

redel(j);

P[P[j].y].val++;

}

}

return 0;

}

int ans[M\*M];//记录选择的行

int change[M\*M][2];//记录每一行是什么意思

int sum;//记录总共多少行

int dfs(){

int next=P[0].r;

if (!next) return 1;

int minval=1234567;

int c=-1;

for(int i=next;i!=0;i=P[i].r){

if(P[i].val<minval){

c=i;

minval = P[i].val;

}

}//每次找该列元素最小的枚举（优化）

modefy1(c);

for(int i=P[c].d;i!=c;i=P[i].d){

for(int j=P[i].r;j!=i;j=P[j].r){

modefy1(P[j].y);

}

ans[0]++;

ans[ans[0]]=P[i].x;

if (dfs()) return 1;

/\*这个顺序很重要，删除和恢复的方向必须相反

开始相同，都是向右的，结果TLE了\*/

ans[0]--;

for (int j=P[i].l;j!=i;j=P[j].l){

modefy2(P[j].y);

}

}

modefy2(c);

return 0;

}

//1-81 每个地方有数字

/\* 1 ... 9

. .

. .

73 81

\*/

//82-162 每列有所有数字

/\*82-90...154-162\*/

//163-243 每行有所有数字

/\*163-171

. .

. .

245-243

\*/

//244-324 每宫有所有数字

/\*244-252 253-261 262-270

271-279 280-288 289-297 .

298-306 307-315 316-324

\*/

int ins[M\*4];

int insert(int x){//x为行号

int leftmost=0;

for(int i=1;i<=ins[0];i++){

pn++;

P[pn].x=x;

P[pn].y=ins[i];//x,y为约束矩阵中坐标

P[ins[i]].val++;//该列元素总数++

P[pn].u=last[ins[i]];

P[last[ins[i]]].d=pn;//上下链接

last[ins[i]]=pn;

if(i==1) leftmost=pn;

if(i==ins[0]) break;

P[pn].r=pn+1;

P[pn+1].l=pn;//左右链接

}

P[leftmost].l=pn;

P[pn].r=leftmost;//最左最右相连接

return 0;

}

int work(int num,int val){

ins[0]=4;

int x=(num-1)/9+1;

int y=(num-1)%9+1;

int k=(x-1)/3\*3+(y-1)/3+1;

ins[1]=num;

ins[2]=81+(y-1)\*9+val;

ins[3]=162+(x-1)\*9+val;

ins[4]=243+(k-1)\*9+val;

insert(++sum);

change[sum][0]=num;

change[sum][1]=val;//记录这行干什么的

return 0;

}

int main(){

freopen("A.in","r",stdin);

while(cin>>s+1){

if(strlen(s+1)<=5)break;

build();

for(int i=1;i<=81;i++){

if(s[i]=='.'){

for(int j=1;j<=9;j++){

work(i,j);

}

}

else {

work(i,s[i]-'0');

}

}

for(int i=1;i<=324;i++){

P[last[i]].d=i;

P[i].u=last[i];//最上最下相连接

}

dfs();

for(int i=1;i<=ans[0];i++){

s2[change[ans[i]][0]]=change[ans[i]][1];

}

for(int i=1;i<=81;i++){

if(s[i]=='.') printf("%d",s2[i]);

else printf("%c",s[i]);

}

printf("\n");

}

return 0;

}