

Koishi 与密码锁

题意简述

有一个 5 位数密码锁。每次操作只能转一个位置或者相邻的两个位置。给出 n 个错误状态，每个错误状态都可以从正确的状态操作一次来达成。求有多少种可能的正确密码。

题目解析

每个位置只有 $0 \sim 9$ 共计 10 种状态，总共只有 10^5 的状态数量，可以考虑枚举。

枚举出每个状态，然后对 n 个状态逐一检查，能否由当前枚举到的状态一步变成。如果能就记入答案。

如此，接下来就是实现的问题了。

代码

```
#include<stdio.h>
#define ll long long

int ans;
int n,inx[10][5];

int roll(int num){
    if(num>=10)
        return num-10;
    else
        return num;
}

char check(int x[]){
    int dif,suc,i,j,k;
    for(i=1;i<=n;i++){
        dif=0;
        for(j=0;j<5;j++){
            if(inx[i][j]!=x[j]){
                dif++;
            }
        }
        //先计算有几位不同，如果完全相同或者有超过两位不同则先否定当前状态
        if(dif>2||dif==0) return 0;
        if(dif==2){ //只有一位不同则一定可以一步到达，如果有两位不同进一步考虑是不是可以通过
            //旋转相邻的两位来到达
            suc=0;
            for(j=0;j<4;j++){ //枚举旋转哪两位
                if(inx[i][j]!=x[j]&&inx[i][j+1]!=x[j+1]){
                    for(k=1;k<10;k++){ //枚举旋转几下，由于是 0~9,因此至多转 9 下
                        if(inx[i][j]==roll(x[j]+k)&&inx[i][j+1]==roll(x[j+1]+k)){
                            suc=1;
                            break;
                        }
                    }
                }
            }
        }
    }
    if(!suc)
```

```

        return 0;
    }
}
return 1;
}

int main(){
    int num[5],i,j;
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        for(j=0;j<5;j++){
            scanf("%d",&inx[i][j]);
        }

        for(num[0]=0;num[0]<=9;num[0]++){
            for(num[1]=0;num[1]<=9;num[1]++){
                for(num[2]=0;num[2]<=9;num[2]++){
                    for(num[3]=0;num[3]<=9;num[3]++){
                        for(num[4]=0;num[4]<=9;num[4]++){
                            if(check(num)) //枚举每一位上的状态，进行检查，如果符合题意就记
入答案
                                ans++;
                        }
                    }
                }
            }
        }
        printf("%d",ans);
        return 0;
    }
}

```

补充

注意到题目中说到密码锁如下图所示，但是图没有放出。然而这题没有图并不影响做题。

当总状态数比较少的时候，可以考虑 **枚举** 全部状态然后逐一检查。这个思想挺常用的。

