题意：给出n次6个数（n<=3），用这n个数能组成1到x的每一个数，输出最大的x（可以不全用）

因为数据范围较小，直接暴力求解，简单的搜索

不可能出现比99大的数字，因为如果要出现比99大的数字，那么需要有1 1，2 2，3 3，4 4，5 5，6 6，7 7，8 8，9 9这样的话已经有18个数字，即使三个骰子也只是刚好18个数字。其中必不可少的是数字0，加上就是19。所以不可能出现比99大的数字。

写法1：

#include <stdio.h>

#include <string.h>

#include<iostream>

#include<cstring>

using namespace std;

int n;

short vis[1010];

short used[10];

short a[10][10];

void dfs(int num,int k)

{

vis[num]=1;

if(k>=n)return;

for(int i=0;i<n;i++)

if(!used[i])

for(int j=0;j<6;j++)

{

used[i]=1;

dfs(num\*10+a[i][j],k+1);

used[i]=0;

}

}

int main()

{

scanf("%d",&n);

for(int i=0;i<n;i++)

for(int j=0;j<6;j++)

scanf("%d",&a[i][j]);

memset(vis,0,sizeof(vis));

for(int i=0;i<=n;i++)

used[i]=0;

dfs(0,0);

int t(0);

while(1)

{

t++;

if(vis[t]==0)

{

printf("%d\n",t-1);

break;

}

}

return 0;

}

写法2：

#include<iostream>

#include<queue>

#include<cstdio>

#include<cstring>

using namespace std;

int main()

{

int n;

int a[10][10];

bool vis[1000];

scanf("%d",&n);

for(int i=0;i<n;i++)

for(int j=0;j<6;j++)

scanf("%d",&a[i][j]);

memset(vis,0,sizeof(vis));

for(int i=0;i<n;i++)

for(int j=0;j<6;j++)

{

vis[a[i][j]]=1;

for(int i1=0;i1<n;i1++)

{

if(i1==i)continue;

for(int j1=0;j1<6;j1++)

vis[a[i][j]\*10+a[i1][j1]]=1;

}

}

int t(0);

while(1)

{

t++;

if(vis[t]==0)

{

printf("%d\n",t-1);

break;

}

}

return 0;

}