**题解**

1. **B题**

**分析：**设x为该球队胜利的场数，y为该球队平局的场数，z为该球队失败的场数，P为该球队总得分，G为球赛总场数。可得

x+y+z=G

3\*x+y=P

这两个式子。然后根据这两个式子来求出将z看为常数的x,y的解

x=(P-G+z)/2;

y=(3\*G-P-3\*z)/2;

然后就遍历z的值（依照题目，需要从大往小的方向遍历），找出满足题目要求的解即可。

**源代码**：

#include <iostream>

using namespace std;

int game[100000], point[100000];

int main()

{

int t;

cin >> t;

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

{

cin >> game[i] >> point[i];

}

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

{

cout<<"Team #"<<j+1<<endl;

cout<<"Games: "<<game[j]<<endl;

cout<<"Points: "<<point[j]<<endl;

cout<<"Possible records:\n";

for (int i = game[j]; i >= 0; i--)

{

int x = (point[j] - game[j] + i) / 2;

int y = (3 \* game[j] - point[j] - 3 \* i) / 2;

//cout<<x<<" "<<y<<" "<<i<<endl;

if (x + y + i == game[j] && 3 \* x + y == point[j]&&x>=0&&y>=0)

cout << x << "-" << y << "-" << i << endl;

}

cout<<endl;

}

//getchar();

//getchar();

return 0;

}

1. **G题**

**分析**：

设两面干净的盘子有x个,一面干净的盘子有y个，两面都不干净的盘子有z个，总共干净的盘子的面数为2\*x+y个

对于普通情况，人可以吃的顿数最多为(2\*x+y)/2+1顿。（x=y=0时需要特判）

而对于特殊情况（就是没有一面干净的盘子的情况）

1. 若z=0，则吃的最多顿数为x+1;
2. 若z>0，则干净的面数需要减少2(包括桌面)，则吃的顿数最多为x；

**源代码：**

**#include<iostream>**

**using namespace std;**

**int veryclear[100000],ce\_clear[100000],unclear[100000];**

**int sum[100000];**

**int main(){**

**int t;**

**cin>>t;**

**for(int i=0;i<t;i++){**

**cin>>veryclear[i]>>ce\_clear[i]>>unclear[i];**

**sum[i]=veryclear[i]\*2+ce\_clear[i];**

**}**

**for(int i=0;i<t;i++){**

**int ans;**

**if(ce\_clear[i]==0){**

**if(unclear[i]==0){**

**ans=(sum[i]==0)?sum[i]:sum[i]/2+1;**

**}**

**else{**

**ans=sum[i]/2;**

**}**

**}**

**else{**

**ans=sum[i]/2+1;**

**}**

**cout<<"Scenario #"<<i+1<<": "<<ans<<endl;**

**}**

**//getchar();**

**//getchar();**

**return 0;**

**}**