**算法训练 新生舞会**

时间限制：1.0s   内存限制：512.0MB

**问题描述**  
　　新生舞会开始了。n名新生每人有三个属性：姓名、学号、性别。其中，姓名用长度不超过20的仅由大小写字母构成的字符串表示，学号用长度不超过10的仅由数字构成的字符串表示，性别用一个大写字符‘F’或‘M’表示。任意两人的姓名、学号均互不相同。换言之，每个人可被其姓名或学号唯一确定。给出m对两人的信息（姓名或学号），判断他们是否能共舞。两人能共舞的充要条件为两人性别相异。  
  
**输入**  
　　第一行一个整数n（2<=n<=1000），表示学生人数。接下来的n行每行依次包含一名新生的姓名、学号、性别，分别用一个空格隔开。  
　　之后的一行是一个整数m(1<=m<=1000)，表示询问的数目。接着的m行每行包含两个信息（姓名或学号），保证两个信息不属于同一人，中间用一个空格隔开。  
  
**输出**  
　　对于每个询问输出一行，如果两人可以共舞，输出一个大写字母‘Y’，否则输出一个大写字母‘N’。  
  
  
**样例输入**  
　　4  
　　John 10 M  
　　Jack 11 M  
　　Kate 20 F  
　　Jim 21 M  
　　3  
　　John 11  
　　20 Jack  
　　Jim Jack  
  
  
**样例输出**  
　　N  
　　Y  
　　N  
  
**提示**  
　　可以把名字和学号都当成字符串处理。可以按以下流程实现。  
  
　　#include<iostream>  
　　#include<cstring>  
　　using namespace std;  
  
　　struct tstudent  
　　{  
　　char name[21];  
　　char num[21];  
　　char sex;  
　　};  
  
　　void readdata(tstudent student[], int n)  
　　{  
**输入N个学生的信息**  
　　}  
  
　　int findstudent(tstudent student[], int n, char\* data)  
　　{  
　　if (data == NULL) return -1;  
  
**判断是否有某个学生的学号或名字等于data，如果有，函数返回该学生在student数组中的序号，否则返回-1**  
　　}  
  
　　void solve(tstudent student[], int n, int m)  
　　{  
　　char x[21], y[21];  
　　for (int i=0; i<m; i++) {  
**输入两个人的信息X、Y。通过调用findstudent函数判断这两个人能否成为舞伴**  
　　}  
　　}  
  
　　int main()  
　　{  
　　int n, m;  
　　tstudent student[1010];  
  
　　cin>>n;  
  
　　readdata(student, n);  
　　cin>>m;  
　　solve(student, n, m);  
　　}

本题的C++参考代码如下：

#include<iostream>

#include<cstring>

using namespace std;

struct tstudent

{

char name[21];

char num[21];

char sex;

};

void readdata(tstudent student[], int n)

{

//输入N个学生的信息

int i;

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

{

cin>>student[i].name>>student[i].num>>student[i].sex;

}

}

int findstudent(tstudent student[], int n, char\* data)

{

if (data == NULL)

return -1;

//判断是否有某个学生的学号或名字等于data，如果有，函数返回该学生在student数组中的序号，否则返回-1

int i;

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

{

if(strcmp(student[i].name,data)==0 || strcmp(student[i].num,data)==0){

// cout<<student[i].name<<"\t"<<student[i].num<<"\t"<<student[i].sex<<"\t"<<data<<"\t"<<strcmp(student[i].name,data)<<"\t"<<strcmp(student[i].num,data)<<endl;

return i;

}

}

return -1;

}

void solve(tstudent student[], int n, int m)

{

char x[21], y[21];

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

//输入两个人的信息X、Y。通过调用findstudent函数判断这两个人能否成为舞伴

cin>>x>>y;

int a = findstudent(student,n,x);

int b = findstudent(student,n,y);

if(student[a].sex == student[b].sex)

cout<<"N"<<endl;

else

cout<<"Y"<<endl;

}

}

int main(void)

{

int n, m;

tstudent student[1010];

cin>>n;

readdata(student, n);

cin>>m;

solve(student, n, m);

return 0;

}

本题的C参考代码如下：

# include <stdio.h>

# include <string.h>

struct stu{

char name[21];

char num[21];

char sex;

};

int main()

{

int i,n,m,j;

char s1[20],s2[10];

char se1,se2;

struct stu stu1[1001];

scanf("%d",&n);

for(i=1;i<=n;i++){

scanf("%s %s %c",&stu1[i].name,&stu1[i].num,&stu1[i].sex);

}

scanf("%d",&m);

for(i=1;i<=m;i++)

{

scanf("%s %s",&s1,&s2);

for(j=1;j<=n;j++)

{

if(strcmp(s1,stu1[j].name)==0 || strcmp(s1,stu1[j].num)==0)

{

se1=stu1[j].sex;

break;

}

}

for(j=1;j<=n;j++)

{

if(strcmp(s2,stu1[j].name)==0 || strcmp(s2,stu1[j].num)==0)

{

se2=stu1[j].sex;

break;

}

}

if(se2==se1) printf("N\n");

else printf("Y\n");

}

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

static Stu[] stu ;

public static void main(String[] args) throws IOException {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int n=Integer.parseInt(br.readLine());

stu=new Stu[n];

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

String []str=br.readLine().split(" ");

Stu tem=new Stu(str[0],str[1],str[2]);

stu[i]=tem;

}

int m=Integer.parseInt(br.readLine());

String[][] str=new String[m][2];

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

String []tem=br.readLine().split(" ");

str[i][0]=tem[0];

str[i][1]=tem[1];

}

String[][] arr=new String[m][2];

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

String tem1=str[i][0];

String tem2=str[i][1];

for(int p=0;p<n;p++){

if(stu[p].getStuID().equals(tem1)||stu[p].getStuName().equals(tem1)){

arr[i][0]=stu[p].getStuSex();

}

if(stu[p].getStuID().equals(tem2)||stu[p].getStuName().equals(tem2)){

arr[i][1]=stu[p].getStuSex();

}

}

}

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

if(!arr[i][0].equals(arr[i][1])){

System.out.println("Y");

}else{

System.out.println("N");

}

}

}

}

class Stu{

String StuID;

String StuName;

String StuSex;

public Stu(String stuID, String stuName, String stuSex) {

super();

StuID = stuID;

StuName = stuName;

StuSex = stuSex;

}

public String getStuID() {

return StuID;

}

public String getStuName() {

return StuName;

}

public String getStuSex() {

return StuSex;

}

}