**判断名次**

问题描述

　　某场比赛过后，你想要知道A~E五个人的排名是什么，于是要求他们每个人说了一句话。（经典的开头……-\_-!）得了第1名的人23，说了假话；得了第5名的人不好意思，也说了假话；为了使求解问题简单，第3名同样说了假话。（奇数名次说假话）

输入格式

　　共5行，各行依次表示A~E说的话。  
　　每行包含一个形如“A>=3”的名次判断，即一个大写字母+关系运算符+一个数字，不包含空格。  
　　大写字母A~E，关系运算<、<=、=、>=、>、!=，数字1~5。注意：等于是“=”不是“==”！

输出格式

　　可能有多解，请按照字典序输出排名序列，每个解一行  
　　最后一行输出解的数量

样例输入

A=2  
D=5  
E>3  
A>2  
B!=1

样例输出

ACDEB  
AECBD  
BADCE  
BCADE  
BDACE  
CEADB  
CEBDA  
7

C++代码：

#include <iostream>

#include <cstring>

#include <cstdio>

using namespace std;

int to[5];

int obj[5];

int par[5];

bool tar[5];

int position[5];

int rec[5];

int cnt;

bool cmp(int opr,int a,int b)

{

if(opr==0)return a>=b;

else if(opr==1)return a<=b;

else if(opr==2)return a==b;

else if(opr==3)return a!=b;

else if(opr==4)return a>b;

else return a<b;

}

void judge()

{

bool tb=true;

for(int i=0;i<5;i++){

if(position[i]%2){

if(!cmp(to[i],position[obj[i]],par[i])){

tb=false;

break;

}

}

else{

if(!cmp(5-to[i],position[obj[i]],par[i])){

tb=false;

break;

}

}

}

if(tb){

printf("%c%c%c%c%c\n",rec[0]+'A',rec[1]+'A',rec[2]+'A',rec[3]+'A',rec[4]+'A');

cnt++;

}

}

void dfs(int num)

{

if(num==5){

judge();

return;

}

for(int i=0;i<5;i++){

if(!tar[i]){

position[i]=num;

rec[num]=i;

tar[i]=true;

dfs(num+1);

tar[i]=false;

}

}

}

int main(int argc, char\*\* argv) {

memset(tar,false,sizeof tar);

for(int i=0;i<5;i++){

char s[5],c1;

scanf("%s",s);

obj[i]=s[0]-'A';

if(strlen(s)==4){

if(s[1]=='<')to[i]=1;

else if(s[1]=='!')to[i]=3;

else to[i]=0;

par[i]=s[3]-'0';

}

else{

if(s[1]=='<')to[i]=5;

else if(s[1]=='=')to[i]=2;

else to[i]=4;

par[i]=s[2]-'0';

}

par[i]-=1;

}

cnt=0;

dfs(0);

printf("%d\n",cnt);

return 0;

}

C代码：

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

int a,b,c,d,e;

int count =0;

char s[5][20];

int check(char s){

if(s=='A') return a;

if(s=='B') return b;

if(s=='C') return c;

if(s=='D') return d;

if(s=='E') return e;

}

int fun(char s[]){ //函数用来判断说话的正误

int i,j;

if(strlen(s)==3){

if(s[1]=='='){

//等于号情况

return check(s[0])==(s[2]-'0');

}else if(s[1]=='>'){

//大于号情况

return check(s[0])>(s[2]-'0');

}else{

//小于号情况

return check(s[0])<(s[2]-'0');

}

}else{

if(s[1]=='>'){

//大于等于

return check(s[0])>=(s[3]-'0');

}else if(s[1]=='!'){

//不等于

return check(s[0])!=(s[3]-'0');

}else{

//小于等于

return check(s[0])<=(s[3]-'0');

}

}

}

int say(int k,int m){

//printf("%d-------%d\n",k,fun(s[m]));

if(k%2==0){

if(fun(s[m])){

return 1;

};

return 0;

}

if(k%2==1){

if(!fun(s[m])){

return 1;

};

return 0;

}

}

int pardon (int a, int b, int c, int d, int e){

if(a==b||a==c||a==d){

return 0;

}

if(a==e||b==c||b==d||b==e||c==d){

return 0;

}

if(c==e||d==e){

return 0;

}

return 1;

}

int main(){

int i,j;

char pai[200][6];

for(i=0;i<5;i++){

gets(s[i]);

}

for(a=1;a<6;a++)

for(b=1;b<6;b++)

for(c=1;c<6;c++)

for(d=1;d<6;d++)

for(e=1;e<6;e++){

//printf("%d%d% d%d%d\n",a,b,c,d,e);

if(!pardon(a,b,c,d,e)){

continue;

}

//printf("%d%d% d%d%d\n",a,b,c,d,e);

if(say(a,0)){

///\*

if(say(b,1)){

if(say(c,2)){

if(say(d,3)){

if(say(e,4)){

//printf("%d%d%d%d%d\n",a,b,c,d,e);

pai[count][a-1]='A';

pai[count][b-1]='B';

pai[count][c-1]='C';

pai[count][d-1]='D';

pai[count][e-1]='E';

pai[count][5]='\0';

//printf("\n");

count++;

//printf("%c%c%c%c%c",a+'A'-1,b+'A'-1,c+'A'-1,d+'A'-1,e+'A'-1);

}

}

}

}

//\*/

//printf("%d----%d\n",a,say(a,0));

//printf("%d%d%d%d%d\n",a,b,c,d,e);

}

}

char temp[5];

for(i=0;i<count;i++){

for(j=i;j<count;j++){

if(strcmp(pai[i],pai[j])>0){

strcpy(temp,pai[i]);

strcpy(pai[i],pai[j]);

strcpy(pai[j],temp);

}

}

}

for(i=0;i<count;i++){

puts(pai[i]);

}

printf("%d\n",count);

system("pause");

return 0;

}

JAVA代码：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

private static pj pj;

private static int count = 0;

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

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

int cs = 0;

String[] pd = new String[5];

String[] zfqj = new String[5];

while (cs < 5) {

pd[cs++] = bf.readLine();

}

pj = new pj(pd);

for (int i = 0; i < 5; i++) {

zfqj[0] = (char) ('A' + i) + "";

for (int j = 0; j < 5; j++) {

if (i != j) {

zfqj[1] = zfqj[0] + (char) ('A' + j);

} else {

continue;

}

for (int k = 0; k < 5; k++) {

if (i != k && j != k) {

zfqj[2] = zfqj[1] + (char) ('A' + k);

} else {

continue;

}

for (int l = 0; l < 5; l++) {

if (i != l && j != l && k != l) {

zfqj[3] = zfqj[2] + (char) ('A' + l);

} else {

continue;

}

for (int m = 0; m < 5; m++) {

if (i != m && j != m && k != m && l != m) {

zfqj[4] = zfqj[3] + (char) ('A' + m);

zx(zfqj[4]);

} else {

continue;

}

}

}

}

}

}

System.out.println(count);

}

private static void zx(String l) {

if (pd(l)) { count++; System.out.println(l); }

}

private static boolean pd(String sr) {

boolean b = true;

for (int i = 0; i < 5; i++) {

if (pj.fh[i] == 0 && (sr.indexOf((char) (i + 'A'))) % 2 == 1 || pj.fh[i] == 3 && (sr.indexOf((char) (i + 'A'))) % 2 == 0) { if (sr.indexOf((char) (pj.zm[i] + 'A')) != pj.sz[i]) { b = false; break; } } else if (pj.fh[i] == 1 && (sr.indexOf((char) (i + 'A'))) % 2 == 1 || pj.fh[i] == -2 && (sr.indexOf((char) (i + 'A'))) % 2 == 0) { if (sr.indexOf((char) (pj.zm[i] + 'A')) <= pj.sz[i]) { b = false; break; } } else if (pj.fh[i] == -1 && (sr.indexOf((char) (i + 'A'))) % 2 != 0 || pj.fh[i] == 2 && (sr.indexOf((char) (i + 'A'))) % 2 == 0) { if (sr.indexOf((char) (pj.zm[i] + 'A')) >= pj.sz[i]) { b = false; break; } } else if (pj.fh[i] == 2 && (sr.indexOf((char) (i + 'A'))) % 2 != 0 || pj.fh[i] == -1 && (sr.indexOf((char) (i + 'A'))) % 2 == 0) {

if (sr.indexOf((char) (pj.zm[i] + 'A')) < pj.sz[i]) {

b = false;

break;

}

} else if (pj.fh[i] == -2 && (sr.indexOf((char) (i + 'A'))) % 2 != 0 || pj.fh[i] == 1 && (sr.indexOf((char) (i + 'A'))) % 2 == 0) {

if (sr.indexOf((char) (pj.zm[i] + 'A')) > pj.sz[i]) {

b = false;

break;

}

} else if (pj.fh[i] == 3 && (sr.indexOf((char) (i + 'A'))) % 2 != 0

|| pj.fh[i] == 0 && (sr.indexOf((char) (i + 'A'))) % 2 == 0) {

if (sr.indexOf((char) (pj.zm[i] + 'A')) == pj.sz[i]) {

b = false;

break;

}

}

}

return b;

}

}

class pj {

int[] zm = new int[5];

int[] fh = new int[5];

int[] sz = new int[5];

String[] jz;

pj(String[] s) {

jz = s;

init();

}

private void init() {

for (int i = 0; i < 5; i++) {

zm[i] = (byte) jz[i].charAt(0) - 'A';

if (jz[i].length() == 3) {

if (jz[i].substring(1, 2).equals("=")) {

fh[i] = 0;

} else if (jz[i].substring(1, 2).equals(">")) {

fh[i] = 1;

}else {

fh[i] = -1;

}

sz[i] = Integer.parseInt("" + jz[i].subSequence(2, 3)) - 1;

} else {

if (jz[i].substring(1, 3).equals("!=")) {

fh[i] = 3;

} else if (jz[i].substring(1, 3).equals(">=")) {

fh[i] = 2;

} else {

fh[i] = -2;

}

sz[i] = Integer.parseInt("" + jz[i].subSequence(3, 4)) - 1;

}

}

}

}