##### **2018年ACM基地招新试题（参考答案）**

##### **提示：以下所给代码均为我提交正确通过的代码，所有题目答案不唯一，我的做法仅供参考。**

**1.2018我们要赢 （10 point(s)）**

#include<cstdio>

int main(){printf("2018\nwe gonna win!\n");

return 0;

}

**2.就不告诉你 （15 point(s)）**

#include<cstdio>

int main(){int m,n,ans,flag;

while(scanf("%d%d",&m,&n)!=EOF){

flag=1;

ans=m\*n;

if(ans==0) printf("0");

while(ans){

if(ans%10==0&&flag==1) ;

else {

printf("%d",ans%10);flag=0;

}

ans=ans/10;

}

printf("\n");

}

}

**测试点分析：**

**测试点1（样例换1个数，3分）：**5 3（输入）51（输出）

**测试点2（一般情况，4分）：**1234 4567（输入）2566007（输出）

**测试点3（答案中包含0，2分）：**4 27（输入）801（输出）

**测试点4（最小位数，1分）：**2 3（输入）6（输出）

**测试点5（极限值，答案首尾包含0，3分）：**10000 10000（输入）1（输出）

**测试点6（一个数是0，2分）：**0 9999（输入）0（输出）

**3.当你老了 （15 point(s)）**

#include<cstdio>

int main(){int x,y,m,n,flag,ans1,ans2;

while(scanf("%d%d%d",&x,&y,&n)!=EOF){

flag=0;

if(x%(n-1)!=0) flag=1;

else m=x/(n-1);

if(m\*n-y<0||m\*n-y>100) flag=1;

else ans1=m\*n-y;

if(m-y<0||m-y>100) flag=1;

else ans2=m-y;

if(flag) printf("Impossible\n");

else printf("%d %d\n",ans1,ans2);

}

return 0;

}

**测试点分析：**

**测试点1（样例换1个数，答案非整数，4分）：**22 6 5（输入）Impossible（输出）

**测试点2（样例换1个数，一般情况，5分）：**21 6 4（输入）22 1（输出）

**测试点3（答案小于0，3分）：**28 5 8（输入）Impossible（输出）

**测试点4（答案大于100，3分）：**100 6 6（输入）Impossible（输出）

**4.新学期寄语 （20 point(s)）**

#include<cstdio>

#include<cstring>

struct student{char name[20];

int number;

int num[15],flag[15];

}stu[10000];

int main(){int i,j,k,count1,count2,m,n;

int forbid[10],flagstu[10000];

while(scanf("%d%d",&m,&n)!=EOF){

count1=count2=0;

memset(flagstu,0,sizeof(flagstu));

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

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

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

scanf("%s %d",stu[i].name,&stu[i].number);

for(j=0;j<stu[i].number;j++){

scanf("%d",&stu[i].num[j]);

for(k=0;k<n;k++){

if(stu[i].num[j]==forbid[k]){

if(!flagstu[i]) count1++;

flagstu[i]=1;

stu[i].flag[j]=1;

count2++;

break;

}

if(k==n-1){

stu[i].flag[j]=0;

}

}

}

}

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

if(flagstu[i]){

printf("%s:",stu[i].name);

for(j=0;j<stu[i].number;j++){

if(stu[i].flag[j])

printf(" %05d",stu[i].num[j]);

}

printf("\n");

}

}

printf("%d %d\n",count1,count2);

}

return 0;

}

**测试点分析：**

**测试点1（样例换1个数，一般情况，9分）：**

4 3

23333 66666 88888

CYLL 3 12345 23456 34567

U 4 99669 66666 88888 66666

GG 2 23333 77777

JJ 3 00012 66666 23333（输入）

U: 66666 88888 66666

GG: 23333

JJ: 66666 23333

3 6（输出）

**测试点2（全部为待查找项，4分）：**

4 2

23333 66666

Yangyan 2 23333 66666

Xiaohong 4 66666 66666 23333 66666

Yuxuan 1 23333

Jiaxiang 3 23333 66666 23333（输入）

Yangyan: 23333 66666

Xiaohong: 66666 66666 23333 66666

Yuxuan: 23333

Jiaxiang: 23333 66666 23333

4 10（输出）

**测试点3（最小样例，0前置，4分）：**

1 1

00000

CJLU 1 00001（输入）

0 0（输出）

**测试点4（最大样例，极限值，3分）：**略（输入）略（输出）

**5.输出CJLU （20 point(s)）**

#include <cstdio>

int n[4];

char c[10005];

int main()

{ int i;

while(scanf("%s",c)!=EOF){

for(i=0;c[i]!='\0';i++){

if(c[i]==67||c[i]==99) n[0]++;

if(c[i]==74||c[i]==106) n[1]++;

if(c[i]==76||c[i]==108) n[2]++;

if(c[i]==85||c[i]==117) n[3]++;

}

for(;n[0]!=0||n[1]!=0||n[2]!=0||n[3]!=0;){

if(n[0]!=0) {printf("C");n[0]--;}

if(n[1]!=0) {printf("J");n[1]--;}

if(n[2]!=0) {printf("L");n[2]--;}

if(n[3]!=0) {printf("U");n[3]--;}

}

printf("\n");

}

return 0;

}

**测试点分析：**

**测试点1（样例稍作调整，一般情况，8分）：**

pCTclnGloRgLrtLhgljkLhGJFaUPewSKgtju（输入）

CJLUCJLUJLLLL（输出）

**测试点2（遍历输出一轮后仅剩下一种字符待输出，3分）：**

pcTlnGloRgLrtLhgljkLhGFauPewSKgt（输入）

CJLULLLLL（输出）

**测试点3（仅剩下一种字符待输出，3分）：**

pTlnGloRgLrtLhglkLhGFaPewSKgt（输入）

LLLLLL（输出）

**测试点4（遍历输出整轮后结束，3分）：**

pCTclnGJoRgLrthUgjkhGFauPewSKgt（输入）

CJLUCJLU（输出）

**测试点5（最大样例，极限值，3分）：**略（输入）略（输出）

**6.交换苹果 （15 point(s)）**

#include<cstdio>

#include<algorithm>

using namespace std;

int apple[100000];

int main()

{ int i,t,n,sum,idea;

while(scanf("%d",&t)!=EOF){

while(t--){

scanf("%d",&n);

for(i=0,sum=0;i<n;i++){

scanf("%d%d",&apple[i],&idea);

sum=sum+idea;

}

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

if(i==0) printf("%d %d\n",apple[n-1],sum);

else printf("%d %d\n",apple[i-1],sum);

}

}

}

return 0;

}

**测试点分析：**

**测试点1（样例稍作调整，一般情况，8分）：**

4

2

0 0

5 30

2

20 25

20 0

3

20 25

20 15

20 10

3

20 25

25 30

30 35（输入）

5 30

0 30

20 25

20 25

20 50

20 50

20 50

30 90

20 90

25 90（输出）

**测试点2（特殊情况，仅有1人参与，无人或部分人有数据存在，4分）：**

3

5

0 0

0 0

0 0

0 0

0 0

1

10 10

4

1 0

0 0

0 1

1 1（输入）

0 0

0 0

0 0

0 0

0 0

10 10

1 2

1 2

0 2

0 2（输出）

**测试点3（最大样例，极限值，3分）：**略（输入）略（输出）

**7.正餐时间 （15 point(s)）**

#include<cstdio>

#include<algorithm>

using namespace std;

struct list{

char c[1100];

int n;

}s[1100],w[1100],d[1100];

int comp(const list &a,const list &b){

return a.n<b.n;

}

int main()

{ int t,appetizer,course,dessert,i,j,k,sum;

while(scanf("%d",&t)!=EOF){

while(t--){

scanf("%d%d%d",&appetizer,&course,&dessert);

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

scanf("%s %d",s[i].c,&s[i].n);

for(j=0;j<course;j++)

scanf("%s %d",w[j].c,&w[j].n);

for(k=0;k<dessert;k++)

scanf("%s %d",d[k].c,&d[k].n);

sort(s,s+appetizer,comp);

sort(w,w+course,comp);

sort(d,d+dessert,comp);

sum=s[appetizer/2].n+w[course/2].n+d[dessert/2].n;

printf("%d %s %s %s\n",sum,s[appetizer/2].c,w[course/2].c,d[dessert/2].c);

}

}

return 0;

}

**测试点分析：**

**测试点1（样例稍作调整，一般情况，6分）：**

2

1 3 2

Fresh\_Cucumber 4

Chow\_Mein 5

Rice\_Served\_with\_Chicken\_Leg 12

Fried\_Vermicelli 17

Steamed\_Dumpling 3

Steamed\_Stuffed\_Bun 4

2 3 1

Stir-fried\_Loofah\_with\_Dried\_Bamboo\_Shoot 33

West\_Lake\_Water\_Shield\_Soup 36

DongPo's\_Braised\_Pork 54

West\_Lake\_Fish\_in\_Vinegar 48

Longjing\_Shrimp 188

DongPo's\_Crisp 18（输入）

20 Fresh\_Cucumber Rice\_Served\_with\_Chicken\_Leg Steamed\_Stuffed\_Bun

108 West\_Lake\_Water\_Shield\_Soup DongPo's\_Braised\_Pork DongPo's\_Crisp（输出）

**测试点2（特殊情况，均只有一组数据，全为偶数或全为奇数，4分）：**

3

1 1 1

Fresh\_Cucumber 4

Chow\_Mein 5

Steamed\_Dumpling 3

2 4 2

Fresh\_Cucumber 14

Stir-fried\_Loofah\_with\_Dried\_Bamboo\_Shoot 33

West\_Lake\_Water\_Shield\_Soup 36

DongPo's\_Braised\_Pork 54

West\_Lake\_Fish\_in\_Vinegar 48

Longjing\_Shrimp 188

DongPo's\_Crisp 18

Steamed\_Stuffed\_Bun 24

3 5 3

Fresh\_Cucumber 34

Stir-fried\_Loofah\_with\_Dried\_Bamboo\_Shoot 33

West\_Lake\_Water\_Shield\_Soup 36

Chow\_Mein 15

Rice\_Served\_with\_Chicken\_Leg 12

Fried\_Vermicelli 47

DongPo's\_Braised\_Pork 54

West\_Lake\_Fish\_in\_Vinegar 38

Steamed\_Dumpling 13

Steamed\_Stuffed\_Bun 14

DongPo's\_Crisp 18（输入）

12 Fresh\_Cucumber Chow\_Mein Steamed\_Dumpling

111 Stir-fried\_Loofah\_with\_Dried\_Bamboo\_Shoot DongPo's\_Braised\_Pork Steamed\_Stuffed\_Bun

86 Fresh\_Cucumber West\_Lake\_Fish\_in\_Vinegar Steamed\_Stuffed\_Bun（输出）

**测试点3，4（最大样例，极限值，5分）：**略（输入）略（输出）

**8.照花前后镜 （15 point(s)）**

#include<cstdio>

#include<cstring>

char c[1000][1000];

int a[1000][1000],b[1000][1000];

int main()

{ int i,j,n,flag;

char p;

while(scanf("%c %d",&p,&n)!=EOF){

getchar();

flag=1;

memset(a,0,sizeof(a));

memset(b,0,sizeof(b));

memset(c,' ',sizeof(c));

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

for(j=0;j<n;j++){

c[i][j]=getchar();

if(c[i][j]!=' ') a[i][j]=1;

else a[i][j]=0;

}

getchar();

}

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

for(j=0;j<n;j++){

b[i][n-j-1]=a[i][j];

}

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

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

if(a[i][j]!=b[i][j]){

flag=0;break;

}

if(flag) printf("bu yao zhao le\n");

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

for(j=0;j<n;j++){

if(b[i][j]) printf("%c",p);

else printf(" ");

}

printf("\n");

}

}

return 0;

}

**测试点分析：**

**测试点1（样例稍作调整，一般情况，6分）：**

& 9

@ @@@@@

@@@ @@@

@ @ @

@@@ @@@

@@@ @@@@@

@@@ @ @ @

@@@ @@@@@

@ @ @ @

@ @@@@@（输入）

&&&&& &

&&& &&&

& & &

&&& &&&

&&&&& &&&

& & & &&&

&&&&& &&&

& & & &

&&&&& & （输出）

**测试点2，3（特殊情况，镜面对称，每个2分，共4分）：**

% 3

@@@

@

@@@（输入）

bu yao zhao le

%%%

%

%%%（输出）

! 4

@@

@ @

@ @

@@

bu yao zhao le

!!

! !

! !

!! （输出）

**测试点4（最小样例，极限值，2分）：**

# 1

@（输入）

bu yao zhao le

#（输出）

**测试点5（最大样例，极限值，3分）：**略（输入）略（输出）

**9.无需调试的游戏程序 （25 point(s)）**

#include<cstdio>

#include<cstring>

int red[5],blue[5],red\_count[5],blue\_count[5];

char Red[5][10]={"iceman","lion","wolf","ninja","dragon"};

char Blue[5][10]={"lion","dragon","ninja","iceman","wolf"};

int initial,red\_init,blue\_init,dragon,ninja,iceman,lion,wolf;

void init(){scanf("%d",&initial);

red\_init=blue\_init=initial;

scanf("%d%d%d%d%d",&dragon,&ninja,&iceman,&lion,&wolf);

red[4]=blue[1]=dragon;

red[3]=blue[2]=ninja;

red[0]=blue[3]=iceman;

red[1]=blue[0]=lion;

red[2]=blue[4]=wolf;

memset(red\_count,0,sizeof(red\_count));

memset(blue\_count,0,sizeof(blue\_count));

}

void print(){

int i1,i2,flag,flag0,flag1,flag2,flaginit1,flaginit2,count1,count2,time1,time2;

flag1=flag2=count1=count2=time1=time2=0;

for(i1=0,i2=0,flaginit1=1,flaginit2=0;flag1<5||flag2<5;){

if(i1==5) i1=0;

if(i2==5) i2=0;

if(flag1<5) flag=0;

if(flag0||flaginit1){

if(flaginit1) flaginit1=0;

if(red\_init>=red[i1]){

flag1=0;

red\_init=red\_init-red[i1];

printf("%03d red %s %d born with strength %d,%d %s in red headquarter\n",time1++,Red[i1],++count1,red[i1],++red\_count[i1],Red[i1]);

flag=1;i1++;

}

else {flag1++;i1++;flaginit1=1;}

if(flag1==5){

printf("%03d red headquarter stops making warriors\n",time1++);

flag=1;

}

}

if(flag2<5) flag0=0;

if(flag||flaginit2){

if(flaginit2) flaginit2=0;

if(blue\_init>=blue[i2]){

flag2=0;

blue\_init=blue\_init-blue[i2];

printf("%03d blue %s %d born with strength %d,%d %s in blue headquarter\n",time2++,Blue[i2],++count2,blue[i2],++blue\_count[i2],Blue[i2]);

flag0=1;i2++;

}

else {flag2++;i2++;flaginit2=1;}

if(flag2==5){

printf("%03d blue headquarter stops making warriors\n",time2++);

flag0=1;

}

}

}

}

int main(){int count,t;

while(scanf("%d",&t)!=EOF){count=0;

while(t--){init();

printf("Case:%d\n",++count);

print();

}

}

return 0;

}

**测试点分析：**

**测试点1（样例稍作调整，红方先于蓝方结束，7分）：**

2

20

3 4 6 6 7

40

3 14 5 6 7（输入）

Case:1

000 red iceman 1 born with strength 6,1 iceman in red headquarter

000 blue lion 1 born with strength 6,1 lion in blue headquarter

001 red lion 2 born with strength 6,1 lion in red headquarter

001 blue dragon 2 born with strength 3,1 dragon in blue headquarter

002 red wolf 3 born with strength 7,1 wolf in red headquarter

002 blue ninja 3 born with strength 4,1 ninja in blue headquarter

003 red headquarter stops making warriors

003 blue iceman 4 born with strength 6,1 iceman in blue headquarter

004 blue headquarter stops making warriors

Case:2

000 red iceman 1 born with strength 5,1 iceman in red headquarter

000 blue lion 1 born with strength 6,1 lion in blue headquarter

001 red lion 2 born with strength 6,1 lion in red headquarter

001 blue dragon 2 born with strength 3,1 dragon in blue headquarter

002 red wolf 3 born with strength 7,1 wolf in red headquarter

002 blue ninja 3 born with strength 14,1 ninja in blue headquarter

003 red ninja 4 born with strength 14,1 ninja in red headquarter

003 blue iceman 4 born with strength 5,1 iceman in blue headquarter

004 red dragon 5 born with strength 3,1 dragon in red headquarter

004 blue wolf 5 born with strength 7,1 wolf in blue headquarter

005 red iceman 6 born with strength 5,2 iceman in red headquarter

005 blue dragon 6 born with strength 3,2 dragon in blue headquarter

006 red headquarter stops making warriors

006 blue headquarter stops making warriors（输出）

**测试点2（红方蓝方同时结束，6分）：**

2

100

10 20 30 40 50

400

200 10 30 100 90（输入）

Case:1

000 red iceman 1 born with strength 30,1 iceman in red headquarter

000 blue lion 1 born with strength 40,1 lion in blue headquarter

001 red lion 2 born with strength 40,1 lion in red headquarter

001 blue dragon 2 born with strength 10,1 dragon in blue headquarter

002 red ninja 3 born with strength 20,1 ninja in red headquarter

002 blue ninja 3 born with strength 20,1 ninja in blue headquarter

003 red dragon 4 born with strength 10,1 dragon in red headquarter

003 blue iceman 4 born with strength 30,1 iceman in blue headquarter

004 red headquarter stops making warriors

004 blue headquarter stops making warriors

Case:2

000 red iceman 1 born with strength 30,1 iceman in red headquarter

000 blue lion 1 born with strength 100,1 lion in blue headquarter

001 red lion 2 born with strength 100,1 lion in red headquarter

001 blue dragon 2 born with strength 200,1 dragon in blue headquarter

002 red wolf 3 born with strength 90,1 wolf in red headquarter

002 blue ninja 3 born with strength 10,1 ninja in blue headquarter

003 red ninja 4 born with strength 10,1 ninja in red headquarter

003 blue iceman 4 born with strength 30,1 iceman in blue headquarter

004 red iceman 5 born with strength 30,2 iceman in red headquarter

004 blue ninja 5 born with strength 10,2 ninja in blue headquarter

005 red lion 6 born with strength 100,2 lion in red headquarter

005 blue iceman 6 born with strength 30,2 iceman in blue headquarter

006 red ninja 7 born with strength 10,2 ninja in red headquarter

006 blue ninja 7 born with strength 10,3 ninja in blue headquarter

007 red iceman 8 born with strength 30,3 iceman in red headquarter

007 blue ninja 8 born with strength 10,4 ninja in blue headquarter

008 red headquarter stops making warriors

008 blue headquarter stops making warriors（输出）

**测试点3（蓝方先于红方结束，数据中出现极大值或极小值，7分）：**

2

400

20 3 300 60 98

1000

332 90 47 8989 83（输入）

Case:1

000 red iceman 1 born with strength 300,1 iceman in red headquarter

000 blue lion 1 born with strength 60,1 lion in blue headquarter

001 red lion 2 born with strength 60,1 lion in red headquarter

001 blue dragon 2 born with strength 20,1 dragon in blue headquarter

002 red ninja 3 born with strength 3,1 ninja in red headquarter

002 blue ninja 3 born with strength 3,1 ninja in blue headquarter

003 red dragon 4 born with strength 20,1 dragon in red headquarter

003 blue iceman 4 born with strength 300,1 iceman in blue headquarter

004 red ninja 5 born with strength 3,2 ninja in red headquarter

004 blue ninja 5 born with strength 3,2 ninja in blue headquarter

005 red ninja 6 born with strength 3,3 ninja in red headquarter

005 blue ninja 6 born with strength 3,3 ninja in blue headquarter

006 red ninja 7 born with strength 3,4 ninja in red headquarter

006 blue ninja 7 born with strength 3,4 ninja in blue headquarter

007 red ninja 8 born with strength 3,5 ninja in red headquarter

007 blue ninja 8 born with strength 3,5 ninja in blue headquarter

008 red ninja 9 born with strength 3,6 ninja in red headquarter

008 blue ninja 9 born with strength 3,6 ninja in blue headquarter

009 red headquarter stops making warriors

009 blue headquarter stops making warriors

Case:2

000 red iceman 1 born with strength 47,1 iceman in red headquarter

000 blue dragon 1 born with strength 332,1 dragon in blue headquarter

001 red wolf 2 born with strength 83,1 wolf in red headquarter

001 blue ninja 2 born with strength 90,1 ninja in blue headquarter

002 red ninja 3 born with strength 90,1 ninja in red headquarter

002 blue iceman 3 born with strength 47,1 iceman in blue headquarter

003 red dragon 4 born with strength 332,1 dragon in red headquarter

003 blue wolf 4 born with strength 83,1 wolf in blue headquarter

004 red iceman 5 born with strength 47,2 iceman in red headquarter

004 blue dragon 5 born with strength 332,2 dragon in blue headquarter

005 red wolf 6 born with strength 83,2 wolf in red headquarter

005 blue ninja 6 born with strength 90,2 ninja in blue headquarter

006 red ninja 7 born with strength 90,2 ninja in red headquarter

006 blue headquarter stops making warriors

007 red iceman 8 born with strength 47,3 iceman in red headquarter

008 red wolf 9 born with strength 83,3 wolf in red headquarter

009 red ninja 10 born with strength 90,3 ninja in red headquarter

010 red headquarter stops making warriors（输出）

**测试点4（红方蓝方均不制造兵力或仅制造一轮，2分）：**

3

0

100 200 300 400 300

30

40 50 60 70 80

100

100 100 100 100 200

（输入）

Case:1

000 red headquarter stops making warriors

000 blue headquarter stops making warriors

Case:2

000 red headquarter stops making warriors

000 blue headquarter stops making warriors

Case:3

000 red iceman 1 born with strength 100,1 iceman in red headquarter

000 blue lion 1 born with strength 100,1 lion in blue headquarter

001 red headquarter stops making warriors

001 blue headquarter stops making warriors（输出）

**测试点5（最大样例，极限值，包括所有情况，3分）：**略（输入）略（输出）

**10.Peak（20 point(s)）**

#include<cstdio>

#include<iostream>

using namespace std;

int num[100001];

int main(){int i,m,n,flag;

while(cin>>m){

while(m--){

flag=1;

cin>>n;

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

cin>>num[i];

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

if(num[i]>=num[i+1])

if(i==0||i==n-1) {flag=0;break;}

else break;

}

for(;i<n-2;i++){

if(num[i]<=num[i+1]) {flag=0;break;}

}

if(flag) printf("Yes\n");

else printf("No\n");

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

num[i]=0;

}

}

return 0;

}

**测试点分析：**

**测试点1（样例稍作调整，一般情况，8分）：**

9

5

1 5 7 3 2

5

1 2 1 2 1

4

1 2 3 4

4

4 3 2 1

3

1 2 1

3

2 1 2

5

1 2 3 1 2

5

0 1 8 7 7

7

9 8 7 6 5 4 3（输入）

Yes

No

No

No

Yes

No

Yes

Yes

No（输出）

**测试点2（最小样例，数组长度很小的情况，3分）：**

6

1

20

2

10 20

2

20 10

3

10 20 10

3

10 10 10

3

20 10 20（输入）

No

No

No

Yes

No

No（输出）

**测试点3（数列全部元素基本单调的情况，3分）：**

10

6

1 2 3 4 5 6

5

5 4 3 2 1

6

2 1 3 4 5 6

5

4 5 3 2 1

6

1 2 3 4 6 5

5

5 4 3 1 2

6

1 2 3 4 5 5

5

5 4 3 2 2

6

1 1 2 3 4 5

5

5 5 4 3 2（输入）

No

No

No

Yes

Yes

No

Yes

No

No

No（输出）

**测试点4（中间出现峰值，数组头尾中间做小改动，3分）：**

7

7

1 2 3 4 3 2 1

8

1 2 3 4 3 2 1 1

8

1 2 3 4 3 2 1 2

7

2 2 3 4 3 2 1

7

3 2 3 4 3 2 1

8

1 2 3 3 4 3 2 1

8

1 2 4 3 4 3 2 1（输入）

Yes

Yes

Yes

No

No

No

No（输出）

**测试点5（最大样例，极限值，3分）：**略（输入）略（输出）