2、

A、永远的聪明王

#include<stdio.h>

#include<stdlib.h> //qsort所需头文件

int cmp(const void \*p1,const void \*p2);

struct Node{

char s[101];

int a;

}node[1000000];

int main(void)

{

int n;

scanf("%d",&n);

int i;

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

scanf("%s %d",node[i].s,&node[i].a);

}

qsort(node,n,sizeof(node[0]),cmp);

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

printf("%s ",node[i].s); //输出相应科目

}

return 0;

}

int cmp(const void \*p1,const void \*p2)

{

struct Node \*m=(struct Node\*)p1;

struct Node \*n=(struct Node\*)p2;

return m->a-n->a; //根据每科所需时间排序

}

C、矩阵求和

#include<stdlib.h>

int cmp(const void \*p1,const void \*p2);

int main(void)

{

int m,n;

scanf("%d%d",&n,&m);

int i,j;

int q;

long long a[2000][2000]={0};

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

for(j=1;j<=m;j++){

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

a[i][j]=a[i-1][j]+a[i][j-1]+a[i][j]-a[i-1][j-1];

}

}

scanf("%d",&q);

while(q--){

int x1,x2,x3,x4,y1,y2,y3,y4;

scanf("%d%d%d%d%d%d%d%d",&x1,&y1,&x2,&y2,&x3,&y3,&x4,&y4);

int num1[4],num2[4];

num1[0]=x1,num1[1]=x2,num1[2]=x3,num1[3]=x4;

num2[0]=y1,num2[1]=y2,num2[2]=y3,num2[3]=y4;

qsort(num1,4,sizeof(int),cmp);

qsort(num2,4,sizeof(int),cmp); //排序后可知相交矩形的坐标

long long sum=0; //sum的类型需要注意

if(x3>x2||y3>y2||x1>x4||y1>y4) //若不相交

sum=a[x2][y2]-(a[x1-1][y2]+a[x2][y1-1])+a[x1-1][y1-1]

+a[x4][y4]-(a[x3-1][y4]+a[x4][y3-1])+a[x3-1][y3-1];

else //相交

sum=a[x2][y2]-(a[x1-1][y2]+a[x2][y1-1])+a[x1-1][y1-1]

+a[x4][y4]-(a[x3-1][y4]+a[x4][y3-1])+a[x3-1][y3-1]

-(a[num1[2]][num2[2]]-(a[num1[2]][num2[1]-1]+a[num1[1]-1][num2[2]])+a[num1[1]-1][num2[1]-1]);

//用num数组存储的坐标获得相交矩形，并减去

printf("%lld\n",sum);

}

return 0;

}

int cmp(const void \*p1,const void \*p2)

{

return \*(int\*)p1-\*(int\*)p2;

}

F、菊花侠大战桃花怪

Hint

若有5堆宝石，菊花侠可以先选择x=4，他拿第四堆的同时也得拿第二堆、第一堆，还剩第三堆和第五堆，桃花怪只能拿第三堆或者第五堆，然后菊花侠拿剩下的一堆，菊花侠拿完，因此菊花侠获胜，输出Win。

此题为博弈论题，可通过反证法证明先手必胜。因此直接输出Win即可

#include<stdio.h>

int main(){

printf("Win");

return 0;

}

I、小植哥哥の好奇

Sample Input 1

5

Sample Output 1

1 4

Hint

小植哥哥悄悄提醒你学妹最多有 2^40个学妹（学妹好多，小植哥哥好高兴（小植哥哥：好耶！~））。

经分析可知，最后站起来的编号一定为平方数（因数有奇数个）

#include<stdio.h>

int main(void)

{

long long n; //此处注意类型

scanf("%lld",&n);

long long i;

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

printf("%lld ",i\*i);

}

return 0;

}

J、Time Management

Sample Input 1

5

2 4

1 12

4 5

7 10

7 8

Sample Output 1

3

思路：将完成时间按先后顺序排序，再判断开始时间是否重叠

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int cmp(const void \*p1,const void \*p2);

struct Node{

int a;//开始时间

int b;//结束时间

}node[50000];

int main(void)

{

int n;

scanf("%d",&n);

int i;

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

scanf("%d%d",&node[i].a,&node[i].b);

}

qsort(node,n,sizeof(node[0]),cmp);

int cnt=0; //最终过题数

int time=0; //时间初始化为0

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

if(node[i].a>=time){

time=node[i].b;

cnt++;

}

}

printf("%d",cnt);

return 0;

}

int cmp(const void \*p1,const void \*p2)

{

struct Node \*m=(struct Node\*)p1;

struct Node \*n=(struct Node\*)p2;

return m->b-n->b; //根据结束时间排序

}

K、小布特烦恼

思维题：n\*(n-1)/2的值为冒泡排序的最大次数，因此直接判断是否为非单调递减

#include<stdio.h>

int main(void)

{

int t;

int a[50000];

int n,i;

scanf("%d",&t);

while(t--){

int flag=0;

scanf("%d",&n);

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

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

}

for(i=0;i<n-1;i++){

if(a[i]<=a[i+1]){

flag=1; //如果有一项为递增，flag=1

break；

}

}

if(flag)

printf("YES\n");

else printf("NO\n"); //别漏回车

}

return 0;

}

L、菜学长的糖糖

思维题：直接对所有数据进行异或运算（两两抵消）

#include<stdio.h>

int main(void)

{

int n;

scanf("%d",&n);

int i,a,sum;

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

scanf("%d",&a);

if(i==0)

sum=a; //这里注意让sum一开始等于第一项

else sum=sum^a;

}

printf("%d",sum);

return 0;

}

M、水题

思维题：经分析，可分为两种情况：

1.n<=k，直接输出n。

2.n>k，如果n-k为奇数，结果为1；反之，结果为0

#include<stdio.h>

int main(void)

{

int n,t,k;

scanf("%d",&t);

while(t--){

scanf("%d%d",&n,&k);

if(n<=k)

printf("%d\n",k-n);

else{

if((n-k)&1){

printf("1\n");

}

else printf("0\n"); //别漏掉回车

}

}

return 0;