4、

第一题

考查结构体排序，

懒人必备 algorithm 头文件中的sort函数，而且默认就是升序，直接上

这里我用c++写的

#include<cstdio>

#include<algorithm>

#include<string>

using namespace std;

struct student

{

int num;

char name[103];

} stu[1000002];

int com(const student& x, const student& y)

{

return x.num < y.num;

}

int main()

{

int n, i;

scanf("%d", &n);

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

{

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

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

}

sort(stu, stu + i,com);

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

{

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

}

return 0;

}

下一题

博弈论，

在经历了好多次尝试以后，得出，不管有几堆，菊花侠总会赢的

所以，直接printf(“WIN”)就行了

#include<stdio.h>

#include<string.h>

int main()

{

int n;

scanf("%d", &n);

printf("Win");

return 0;

}

下一题

题目告诉，只能交换i与i-1个，

这个和冒泡排序的方式相同，如果，每两个都要交换，那么

交换的次数就是（n-1)\*n/2，

题目又说，若次数小于（n-1)\*n/2-1次才输出YES，那么就是，当他每两个都要交换的时候，输出NO，

所以，我们只要和冒泡一样，两两比较，在判断是不是全部都要交换即可

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int a[1000][50000];

int g[1000];

int main()

{

int times,nums,c=0;

scanf("%d", &times);

for (int i = 1; i <= times; i++)

{

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

for (int j = 1; j <= g[i]; j++)

{

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

}

}

for (int i = 1; i <= times; i++)

{

c = 0;

for (int j = 1; j < g[i]; j++)

{

if (a[i][j] > a[i][j+1])

c++;

}

if (c == g[i] - 1)

{

printf("NO\n");

}

else

{

printf("YES\n");

}

}

return 0;

}

下一题

众所周知，两个相同的数异或，就是0,0与任何数异或都是他本身，

那么，题目已知，有n个数都是成对的，让他们全部异或以后就是0，这个0再与那个单独的数异或，就得到这个数本身，

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int a;

int main()

{

int x,n;

scanf("%d", &x);

for (int i = 1; i <= x; i++)

{

scanf("%d", &n);

a ^= n;

}

printf("%d", a);

return 0;

}

下一题

注意看，这个题的时间限制长达6秒，所以，我们可以直接枚举出来，

定义一个与矩阵相同的二维数组来标记这个点是否已经被询问过，

要是在第一次已经被询问过，那么第二次询问的时候，就可以直接跳过

为了防爆，尽量定义的时候大一点。

#include<stdio.h>

#include<string.h>

int num[1002][1002];

int a[1000000][9], book[1002][1002] = { 0 };

int main()

{

long long N, M, Q, sum=0;

scanf("%lld%lld", &N, &M);

for (int i = 1; i <= N; i++)

{

for (int j = 1; j <= M; j++)

{

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

}

}

scanf("%lld", &Q);

for (int k = 1; k <= Q; k++)

{

scanf("%d%d%d%d%d%d%d%d", &a[k][1], &a[k][2], &a[k][3], &a[k][4], &a[k][5], &a[k][6], &a[k][7], &a[k][8]);

}

for (int k = 0; k < Q; k++)

{

memset(book, 0, sizeof(book));

for (int i = a[k + 1][1]; i <= a[k + 1][3]; i++)

{

for (int j = a[k + 1][2]; j <= a[k + 1][4]; j++)

{

if (book[i][j] == 0)

{

sum += num[i][j];

book[i][j] = 1;

}

}

}

for (int i = a[k + 1][5]; i <= a[k + 1][7]; i++)

{

for (int j = a[k + 1][6]; j <= a[k + 1][8]; j++)

{

if (book[i][j] == 0)

{

sum += num[i][j];

book[i][j] = 1;

}

}

}

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

}

return 0;

}

下一题

名字叫水题，为啥我觉得一点也不水（飘过）（飘过）

又要找规律，一共大致分为三种情况

kn,k<n,k>n

当kn的时候，这个b点就在2n处

当k>n的时候，此时b点肯定在a点的右边，但是不能取到，所以，得把a点挪到k这个位置，让其刚好成为上一种情况，挪动的距离就是k-n

当k<n的时候，此时b点在a点的左侧，此时，如果他们的奇偶性相同才能取到，如果不同的话，得先挪动，将他们的奇偶性变成相同的，即挪动一个点

#include<stdio.h>

int main()

{

int t,n,k;;

scanf("%d",&t);

while(t--)

{

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

if(n==k)

printf("0\n");

else if(n>k)

{

if((n-k)%2==0)

printf("0\n");

else

printf("1\n");

}

else

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

}

return 0;

}

下一题

又到了最喜欢的找规律环节；

咋们可以先按照他说的那样，在编译器上试一下

当我们输入5的时候，结果是1,4.

好像看不出来啥，那就换个大一点的

输入100；

结果很奇妙：1,4,9,16,25,36,49,64,81…

amazing，他就是数的平方

所以，直接枚举就可以

#include<stdio.h>

int main()

{

int n,i=1,t=1;

scanf("%d", &n);

while (i <= n)

{

printf("%d ", i);

i = t \* t;

i++;

}

return 0;

}