1. 反向输出整数

将一个三位的整数反向输出，例如127，反向输出为721

eg1：

970

079

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int n,a,b,c;

    cin>>n;

    a=n/100;

    b=(n/10)%10;

    c=n%10;

    printf("%d%d%d",c,b,a);

    return 0;

}

1. 阶乘计算

计算输入的阶乘，如果无法计算则返回0

eg1.

3

6

eg2.

10

3628800

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int N;

    cin>>N;

    int res=1;

    if(N<=0){res=0;cout<<res;}

    else{

        for(int i=1;i<=N;++i){

            res\*=i;

        }

        cout<<res;

    }

    return 0;

}

1. 依依的菜地

依依有一块长方形的菜地，长a米，宽b米，菜地的外圈环绕一条c米宽的道路，求这条环道的面积。

eg1.

8 5 1

30

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int a,b,c;

    scanf("%d %d %d",&a,&b,&c);

    int area;

    area=(b+2\*c)\*(a+2\*c)-a\*b;

    cout<<area;

    return 0;

}

1. 摸黑过桥

度度熊和他的朋友们（人数为n）去野外露营，很快天黑了，他们需要过一座桥，桥很窄，每次最多允许两人通过，但是他们只有1个手电筒，所以每次过完桥，需要有人带回手电筒，设第i号小朋友的过桥时间为T[i]，两人过桥的时间为耗时最长者，问所有人过桥的最短耗时是多少？假设n为10，且他们每个人的耗时为：{1,2,5,10,12,23,43,50,120,122}

eg1.

4

17

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int n;

    cin>>n;

    int minTime=0;

    //两者中耗时长的人不让他回来送电灯

    //n-1次过桥

    //让最小的一直送电灯？

    int arr[10]={1,2,5,10,12,23,43,50,120,122};

    for(int i=n-1;i>0;--i){

        minTime+=arr[i];

    }

    cout<<minTime;

    return 0;

}

1. 乘积最大化

度度熊和朋友在玩一个游戏：给定一个数n，将数n拆成m个数（要求m个数的和等于n)：num1,num2...，其中m>1,m<=n，请问num1 num2...这m个数的最大乘积是多少?

例如:

n于5，m可以等于2，此时2个数分别为：2 3，得到最大乘积为6。(此处两个数也可以取1和4，但是乘积为4，不是最大的)

n等于9，m可以等于3，此时3个数分别为：3 3 3，得到最大乘积为27

eg1.

5

6

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int n;

    cin>>n;

    if(n==1)cout<<1<<endl;

    else if(n==2)cout<<1<<endl;

    else if(n==3)cout<<2<<endl;

    else if(n==4)cout<<4<<endl;

    else{

        int RES[n+1];

    RES[0]=0;RES[1]=1;RES[2]=2;RES[3]=3;RES[4]=4;

    for(int i=5;i<=n;++i){

        int tmp=0;

        for(int j=1;j<=i-1;++j){

            if(j\*RES[i-j]>tmp)tmp=j\*RES[i-j];

        }

        RES[i]=tmp;

    }

    cout<<RES[n];

    }

    return 0;

}

1. 配对游戏(easy)

给你两个由正整数组成且长度相等的序列(a},{b}，对于序列{a}中的每个元素a[i]，你需要选择序列{b}中的某个元素b[j]与之配对。序列{b}中的每个元素必须和序列{a}中某一元素配对，且仅能配对一次。对于这n组数对(a[i],b[j])，如果恰好有k组数对满足：a[i]>b[j],那么你将获得456亿元的奖金，那么你有多少种方案能获得奖金呢?

输入格式：输入包含三行

第一行包含两个正整数n(1≤n≤10),代表序列{a},{b}的长度,k(0≤k≤n),含义如题面所述

第二行包含n个正整数a1,a2, ... ,an (1≤ai≤10^9)

第二行包含n个正整数b1,b2, ... ,bn (1≤bi≤10^9)

数据保证：不存在ai=aj(i≠j)且不存在bi=bj(i≠j)且不存在ai=bj(1≤i,j≤n)

输出格式：输出一个非负整数，含义如题面所述

eg1.

4 3

1 6 3 8

7 4 2 5

4

备注：

对于第一个测试数据，共有如下4种方案:

{(1,7)、(6,4)、(3, 2)、(8,5)}

{(1,7)、(8,4)、(3,2)、(6,5)}

{(8,7)、(1,4)、(3,2)、(6,5)}

{(8,7)、(6,4)、(3,2)、(1,5)}

如果两个方案称之为不同，那么至少存在一个序列{a}中的元素，在不同方案中与序列{b}中不同元素匹配

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int n;

    cin>>n;

    int a[n],b[n];

    int res=0;

    for(int i=0;i<n;i++){

        cin>>a[i];

    }

    for(int i=0;i<n;i++){

        cin>>b[i];

    }

    cout<<res;

    return 0;

}

1. 配对游戏(hard)

给你两个由正整数组成且长度相等的序列{a},{b},对于序列{a}中的每个元素a[i],你需要选择序列(b}中的某个元素b[j]与之配对,序列{b}中的每个元素必须和序列{a}中某一元素配对，且仅能配对一次,对于这n组数对(a[i],b[j])，如果恰好有k组数对满足：a[i]>b[j],那么你将获得456亿元的奖金。那么你有多少种方案能获得奖金呢？答案可能很大，请输出答案对1000000007（10^9+7)取模后的结果

输入格式：输入包含三行

第一行包含两个正整数n(1≤n≤10),代表序列{a},{b}的长度,k(0≤k≤n),含义如题面所述

第二行包含n个正整数a1,a2, ... ,an (1≤ai≤10^9)

第二行包含n个正整数b1,b2, ... ,bn (1≤bi≤10^9)

数据保证：不存在ai=aj(i≠j)且不存在bi=bj(i≠j)且不存在ai=bj(1≤i,j≤n)

输出格式：输出一个非负整数，含义如题面所述

eg1.

4 3

1 6 3 8

7 4 2 5

4

备注：

对于第一个测试数据，共有如下4种方案:

{(1,7)、(6,4)、(3, 2)、(8,5)}

{(1,7)、(8,4)、(3,2)、(6,5)}

{(8,7)、(1,4)、(3,2)、(6,5)}

{(8,7)、(6,4)、(3,2)、(1,5)}

如果两个方案称之为不同，那么至少存在一个序列{a}中的元素，在不同方案中与序列{b}中不同元素匹配

#include<bits/stdc++.h>

using namespace std;

int main( ){

    int n;

    cin>>n;

    int a[n],b[n];

    int res=0;

    for(int i=0;i<n;i++){

        cin>>a[i];

    }

    for(int i=0;i<n;i++){

        cin>>b[i];

    }

    cout<<res;

    return 0;

}