列出所有的可能性 新知一下

逻辑航线

暴力枚举（初级）

NOI基础算法系列课程

版本：3.0.0

讲师：孙伟航

01

简单枚举

例1-水仙花数

逻辑航线

题目描述

水仙花数是指一个3位数,它的每个位上的数字的3次幂之和等于它本身。例如13+53+33=1531^3+5^3+3^3=15313+53+33=153.

输入格式

无

输出格式

每行一个水仙花数

题目地址：https://www.luogu.com.cn/problem/U69290

例1-解题思路

枚举范围：100-999。
枚举条件：个位的三次方+十位的三次方+百位的三次方等于枚举数。

例1-AC代码

逻辑航线

```
#include <iostream>
using namespace std;
int main(int argc, char** argv)
{
    for(int i=100; i<=999; i++)
    {
        int a = i % 10;
        int b = i % 100 /10;
        int c = i / 100;
        if(a*a*a + b*b*b + c*c*c == i)
        {
                cout << i << endl;
        }
    }
    return 0;
}
```

例2-烤鸡

逻辑航线

题目描述

猪猪 Hanke 特别喜欢吃烤鸡（本是同畜牲，相煎何太急！）Hanke 吃鸡很特别，为什么特别呢？因为他有 10 种配料（芥末、孜然等），每种配料可以放 1 到 3克，任意烤鸡的美味程度为所有配料质量之和。

现在， Hanke 想要知道，如果给你一个美味程度 n ， 请输出这 10种配料的所有搭配方案。

输入格式

一个正整数 n，表示美味程度。

输出格式

第一行，方案总数。
第二行至结束，10 个数，表示每种配料所放的质量，按字典序排列。
如果没有符合要求的方法，就只要在第一行输出一个 0。

题目地址：https://www.luogu.com.cn/problem/P2089

例2-解题思路1

枚举范围：共10种调料，每种可以使用1-3克
枚举条件：10种调料的重量和等于输入目标重量

例2-AC代码

逻辑航线

```
#include <iostream>
#include <cstdio>
using namespace std;
#define rep(i,a,b) for (int i = a; i <= b; i++)
int main(int argc, char** argv)
{
    int n,ans = 0,cnt =10;
    scanf("%d",&n);
    rep(a,1,3)rep(b,1,3)rep(c,1,3)rep(d,1,3)rep(e,1,3)rep(f,1,3)rep(g,1,3)rep(h,1,3)rep(i,1,3)rep(j,1,3)
    if(a+b+c+d+e+f+g+h+i+j == n)
        ans++;
    printf("%d\n",ans);
    rep(a,1,3)rep(b,1,3)rep(c,1,3)rep(d,1,3)rep(e,1,3)rep(f,1,3)rep(g,1,3)rep(h,1,3)rep(i,1,3)rep(j,1,3)
    if(a+b+c+d+e+f+g+h+i+j == n)
        printf("%d %d %d %d %d %d %d %d %d %d\n",a,b,c,d,e,f,g,h,i,j);
    return 0;
}
```

例2-解题思路2

如何优化，减少枚举数量？

例如：A+B = 4，A、B的取值范围分别为1到3。求AB组合的可能性。

分析，这个时候我们还需要分别枚举A、B吗？

答案是**不需要**。因为，当A确定的时候， B的取值等于4减A，是唯一确定的。当A分别等于1,2,3的时候，B分别等于3,2,1，无需枚举B，这就大大减少了枚举量。

例2-解题思路2

回到本题，我们先来确定一下第一种调料A的取值。它存在以下4种情况：

情况1：美味值为30时，A只有一种可能，即等于3克，可以看成3到3。
情况2：美味值为29时，A有两种可能，即2克或者3克，可以看成2到3。
情况3：美味值小于等于28且大于10时，A存在三种可能，即：1克、2克或者3，可以看成1到3。
情况4：美味值等于10时，A只有一种可能，即等于1克，可以看成1到1。

所以我们能够根据美味值，来提前锁定A的取值范围，即**max(1,n-27)到min(3,n-9)**。当A的值锁定的时候，我们就可以来推导B的取值，即**max(1,n-24-A)到min(3,n-8-A)**。

以此类推：

C = max(1,n-21-A-B)到min(3,n-7-A-B)，
D = max(1,n-21-A-B-C)到min(3,n-7-A-B-C)，
E = max(1,n-18-A-B-C-D)到min(3,n-6-A-B-C-D)
……
I = max(1,n-3-A-B-C-D-E-F-G-H)到min(3,n-1-A-B-C-D-E-F-G-H)
J= max(1,n-A-B-C-D-E-F-G-H-I)到min(3,n-A-B-C-D-E-F-G-H-I)

例2-AC代码

逻辑航线

```
#include <iostream>
#include <algorithm>
using namespace std;
#define rep(i,a,b) for (int i = max(1,a); i <= min(3,b); i++)
int li[60000][10];
int main(int argc, char** argv)
{
	int n,ans = 0,cnt =10;
	scanf("%d",&n);
	rep(a,n-27,n-9)rep(b,n-24-a,n-8-a)rep(c,n-21-a-b,n-7-a-b)rep(d,n-18-a-b-c,n-6-a-b-c)
	rep(e,n-15-a-b-c-d,n-5-a-b-c-d)rep(f,n-12-a-b-c-d-e,n-4-a-b-c-d-e)rep(g,n-9-a-b-c-d-e-f,n-
	3-a-b-c-d-e-f)rep(h,n-6-a-b-c-d-e-f-g,n-2-a-b-c-d-e-f-g)rep(i,n-3-a-b-c-d-e-f-g-h,n-1-a-b-
	c-d-e-f-g-h)rep(j,n-a-b-c-d-e-f-g-h-i,n-a-b-c-d-e-f-g-h-i)
	{
		li[ans][0]=a;li[ans][1]=b;li[ans][2]=c;li[ans][3]=d;li[ans][4]=e;
		li[ans][5]=f;li[ans][6]=g;li[ans][7]=h;li[ans][8]=i;li[ans][9]=j;
		ans++;
	}
	printf("%d\n",ans);
	for(int i=0; i<ans; i++)
	{
		for(int j=0; j<10; j++)
			printf("%d ",li[i][j]);
		printf("\n");
	}
	return 0;
}
```

例3-三连击(升级版)

逻辑航线

题目描述

将 1,2,…9共 9个数分成三组，分别组成三个三位数，且使这三个三位数的比例是A:B:C，试求出所有满足条件的三个三位数，若无解，输出No!!!

输入格式

三个数，A、B、C。

输出格式

若干行，每行 3 个数字。按照每行第一个数字升序排列。

题目地址：https://www.luogu.com.cn/problem/P1618

例3-解题思路

完全暴力枚举，将9个数字分别从1-9进行尝试，最终将达到9^9次计算，无法在一秒内完成。所以不可使用。

解析题目，我们能够推导出两个关键信息：
枚举范围： 123到987。
枚举对象：仅需**第一个三位数**，对于后面两个数字，我们可以根据比例进行计算得出。

例如第一个数字为123，输入的比例为1,2,3，那么后面的两个三位数就应该分别是246,369，又因为这3个三位数中存在大量的重复数字，如2,3,6，所以该组合不满足题意。

此题目中还包含第二个问题，如何判断1-9这九个数字完全被使用而没有重复？很简单，我们仅需要使用一个长度为10的数组桶，来对已使用的数字进行标记即可。

比例小知识：A：B = 2:5，已知A=3，求B?
B = 5 * A / 2

例3-AC代码


```cpp
#include <iostream>
#include <algorithm>
#include <cstdio>
#include <cstring>
using namespace std;
int b[10];
//分解3位数的各个数字，放在桶中标记为已使用。
void markNum(int x)
{
	b[x % 10] = 1;
	b[x / 10 % 10] = 1;
	b[x / 100] = 1;
}

bool check(int x,int y,int z)
{
	memset(b,0,sizeof(b));
	if(y>999 || z > 999) return false;
	markNum(x),markNum(y),markNum(z);
	for(int i=1; i<=9; i++)
	if(!b[i])return false; //有的数字没有被使用，所以不符合题意。
	return true;
}
```

```cpp
int main(int argc, char** argv)
{
	long long A,B,C,x,y,z,cnt = 0;
	cin >> A >> B >> C;
	for(x=123; x<=987; x++)
	{
		if(x * B % A || x * C % A ) continue; //彼此之间不成比例，跳入下一个循环
		y = x * B / A;
		z = x * C / A;
		if(check(x,y,z))
		{
			printf("%lld %lld %lld\n",x,y,z);
			cnt++;
		}
	}
	if(cnt == 0)
		puts("No!!!");
	return 0;
}
```

例4-统计方形（加强版）

逻辑航线

题目描述

有一个 n×m 方格的棋盘，求其方格包含多少正方形、长方形（不包含正方形）。

输入格式

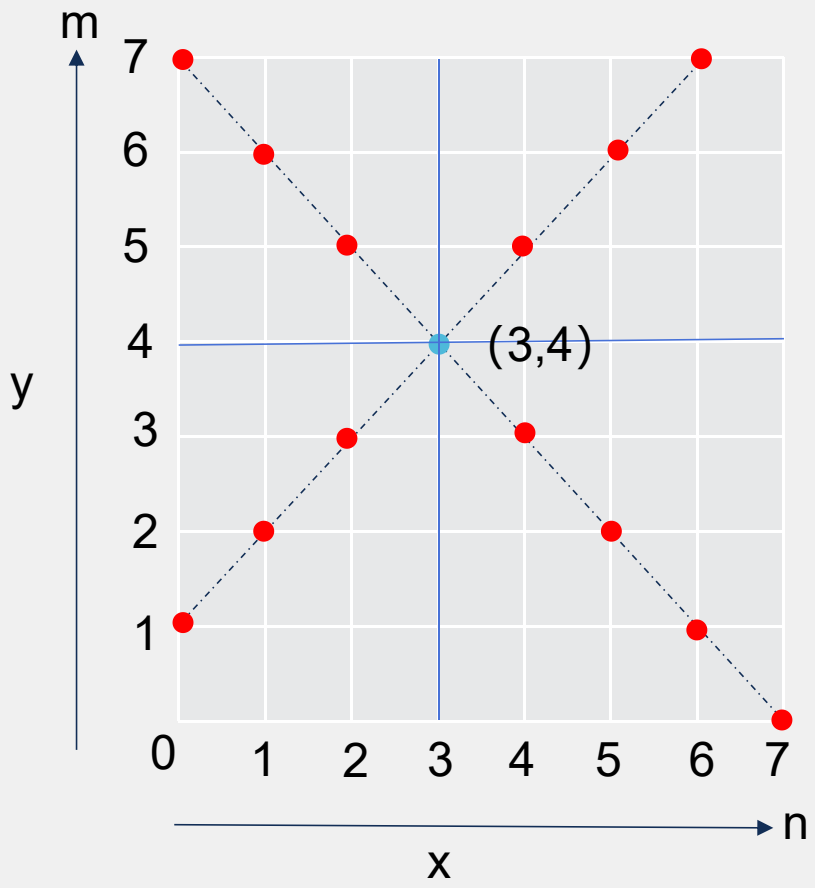
一行，两个正整数 n,m（n≤5000,m≤5000 ）。

输出格式

一行，两个正整数，分别表示方格包含多少正方形、长方形（不包含正方形）。

题目地址：https://www.luogu.com.cn/problem/P2241

例4-解题思路1



以3,4点为例，我们可以发现，图上的每一个红点都能与它组成正方形，即图中的红点数就是以该点作为顶点的正方形数量。

剩下的，除了两条蓝线上的点，都能够与3,4组成长方形。

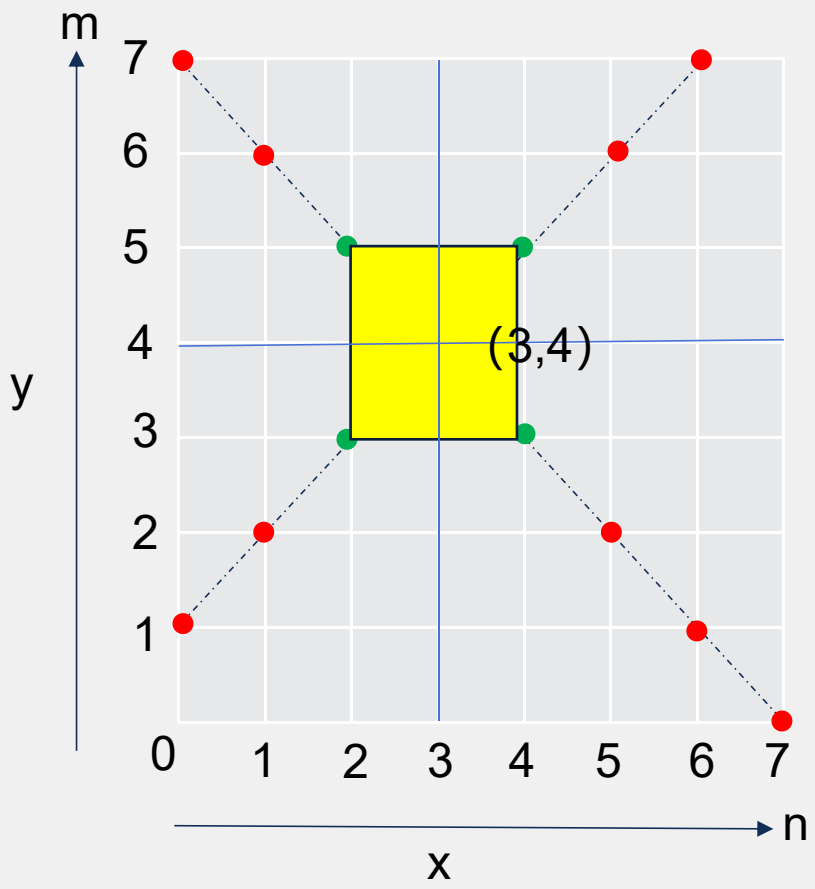
总点数 = m*n（排除了两条蓝线上的点）。

指定点左上角的红点数A：min(x,m-y)
指定点左下角的红点数B：min(x,y)
指定点右上角的红点数C：min(n-x,m-y)
指定点右下角的红点数D：min(n-x,y)

指定点正方形总数 = A+B+C+D
指定点长方形总数 = 总点数 – 指定点正方形总数

但是，答案正确吗?

例4-解题思路1的BUG



黄色区域的正方形分别被四个绿色的顶点各自计算了一次，导致最终结果是正确答案的四倍！

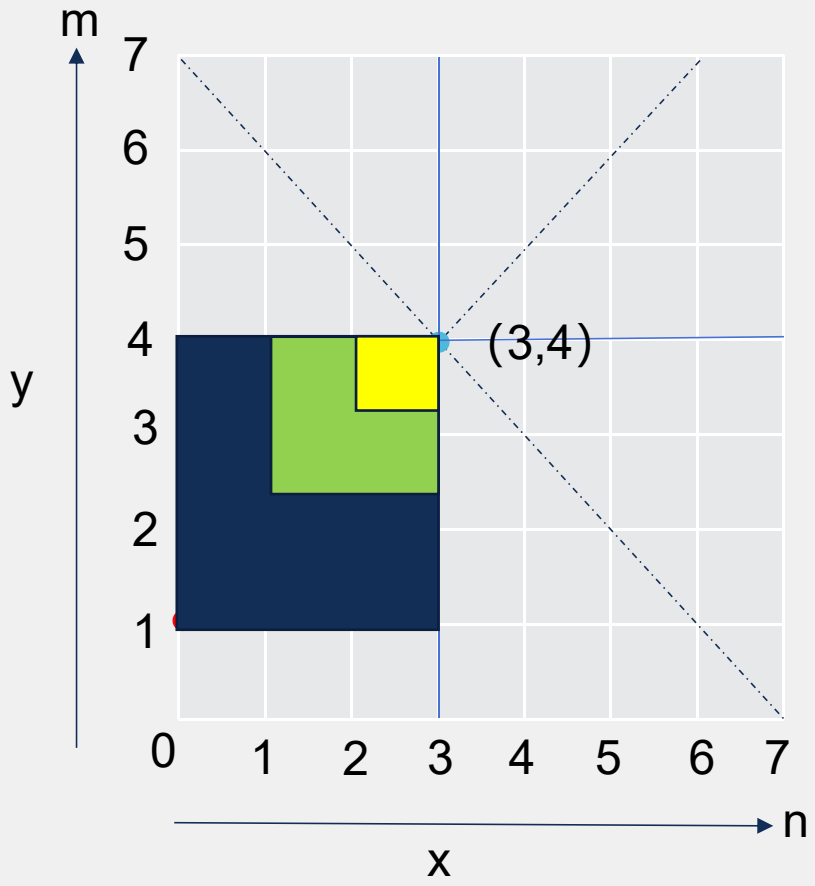
因此，我们要将统计之后的正方形与长方形数量各自除以4。

例4-AC代码

逻辑航线

```
#include <algorithm>
#include <cstdio>
using namespace std;
typedef long long LL; //把long long 替换成LL，节约录入时间
int main(int argc, char** argv)
{
    LL n,m, squ = 0, rec = 0;
    scanf("%lld%lld",&n,&m);
    for(LL x=0; x<=n; x++)
    {
        for(LL y=0; y<=m; y++)
        {
            LL tmp = min(x,y) + min(y,n-x) + min(n-x,m-y) + min(m-y,x);
            squ += tmp;
            rec += n * m - tmp;
        }
    }
    printf("%lld %lld",squ / 4, rec / 4); //修正最后答案
    return 0;
}
```

例4-解题思路2



思路1的方案存在大量的冗余，我们能不能只统计右上角呢?

我们现在可以看出，点3,4分别是黄、绿、蓝三个正方形的右上角。并且，按照右上角的方式进行统计则不会出现任何的重复。

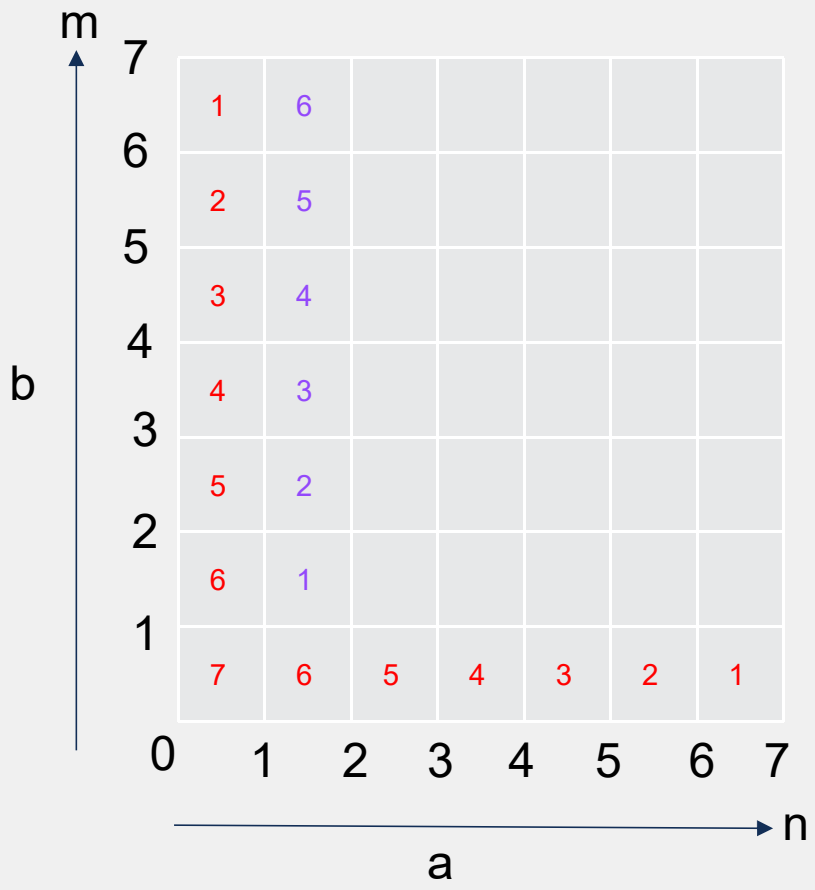
但是注意，此时的长方形的数量，我们就不能再使用n*m来减掉当前点的正方形数量了，而应该使用x * y – 当前点的正方形数量。

例4-AC代码


```cpp
#include <algorithm>
#include <cstdio>
using namespace std;
typedef long long LL; //把long long 替换成LL，节约录入时间

int main(int argc, char** argv)
{
    LL n,m, squ = 0, rec = 0;
    scanf("%lld%lld",&n,&m);
    for(LL x=0; x<=n; x++)
    {
        for(LL y=0; y<=m; y++)
        {
            LL tmp = min(x,y);
            squ += tmp;
            rec += x * y - tmp;
        }
    }
    printf("%lld %lld",squ, rec);
    return 0;
}
```

例4-解题思路3



更换枚举条件。

考虑n*m大小的矩形中，能够包含多少个a*b大小的矩形?

在7*7矩阵中，

当 a = 1且b = 1时，总共包含7 * 7 = 49个矩形，红色字体。
当 a = 1且b = 2时，总共包含7 * 6 = 42个矩形，紫色字体。

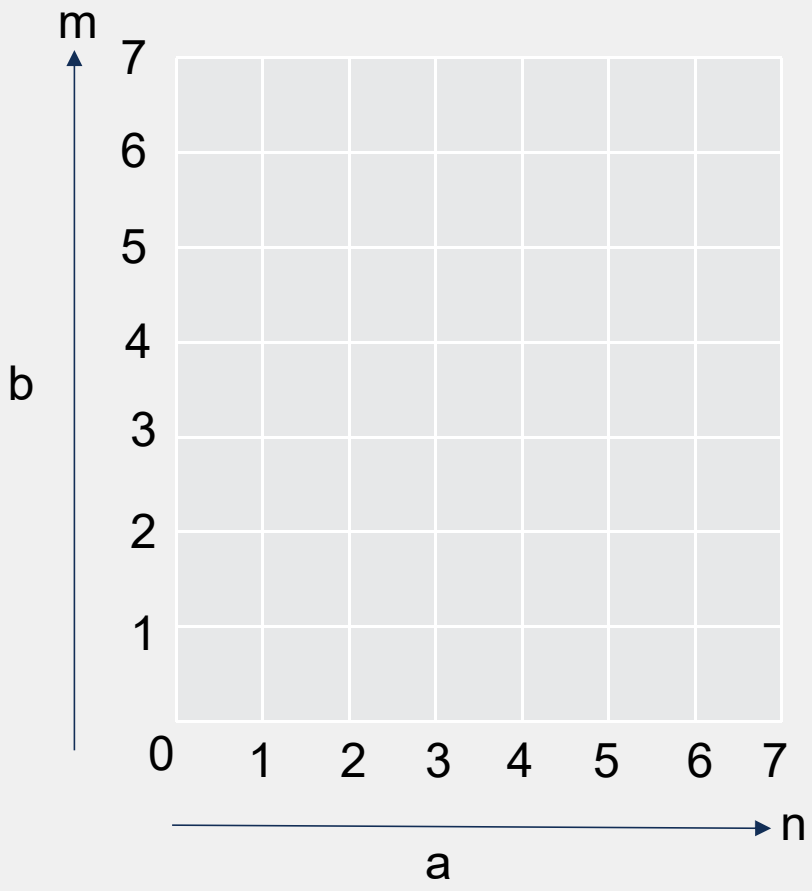
当a、b变化时，可以推出横轴可能的数量 = n-a+1;
纵轴可能的数量 = m – b +1;
当 a == b时，为正方形；其余为长方形。

例4-AC代码

逻辑航线

```cpp
#include <algorithm>
#include <cstdio>
using namespace std;
typedef long long LL; //把long long 替换成LL，节约录入时间
int main(int argc, char** argv){
    LL n,m, squ = 0, rec = 0;
    scanf("%lld%lld",&n,&m);
    for(LL x=1; x<=n; x++){
        for(LL y=1; y<=m; y++){
            if(x==y){
                squ += (n - x + 1) * (m - y + 1);
            }
            else{
                rec += (n - x + 1) * (m - y + 1);
            }
        }
    }
    printf("%lld %lld",squ, rec);
    return 0;
}
```

例4-解题思路4



继续更换枚举的条件，能不能只枚举一个条件?

通过上面的分析，我们可以得出，当a=1时，n方向上边长共有7种可能性，当a=2时，n方向上边长共有6种可能性……当a=7时，n方向上边长只有一种可能性。

所以，n方向上总共的边长可能性是N = 1+2+3+……n = (1+n)n/2。
同理，m方向上总共的边长可能性是M = (1+m)m/2。

N * M就是总共的矩形数量。

那么，我们只需要枚举一条边长，就能求出正方形的数量，再用总的矩形数量减去正方形的数量，即可求出长方形的数量。

例4-AC代码

逻辑航线

```
#include <algorithm>
#include <cstdio>
using namespace std;
typedef long long LL; //把long long 替换成LL，节约录入时间

int main(int argc, char** argv)
{
    LL n,m, squ = 0, rec = 0;
    scanf("%lld%lld",&n,&m);
    for(LL x=1; x<=min(m,n); x++)
    {
        squ += (n - x + 1) * (m - x + 1);
    }
    rec = n * (n + 1) * m * (m + 1) / 4 - squ;
    printf("%lld %lld",squ, rec);
    return 0;
}
```

例5-滑雪课程设计

逻辑航线

题目描述

农民约翰的农场里有 n座山峰，每座山都有一个在 0到 100之间的整数的海拔高度。在冬天，因为山上有丰富的积雪，约翰经常开办滑雪训练营。

不幸的是，约翰刚刚得知税法在滑雪训练营方面有新变化，明年开始实施。在仔细阅读法律后，他发现如果滑雪训练营的最高和最低的山峰海拔高度差大于 17就要收税。因此，如果他改变山峰的高度（使最高与最低的山峰海拔高度差不超过 17 ），约翰可以避免支付税收。

如果改变一座山 x 单位的高度成本是 X^2单位，约翰最少需要付多少钱才能使海拔最高的山峰与海拔最低的山峰的高度只差不超过 17 约翰只愿意改变整数单位的高度。

输入格式

输入的第一行是一个整数，代表山峰的数量 n。
第 2行到(n+1)行，每行一个整数。第 i行的整数 ai代表第 i座山的海拔高度。

输出格式

输出一行一个整数，代表约翰需要支付修改山海拔高度的总金额。

题目地址：https://www.luogu.com.cn/problem/P3650

例5-解题思路

枚举山峰的最小高度，判断总花费即可

例5-AC代码

逻辑航线

```
#include <iostream>
#include <cstdio>
using namespace std;
int a[1005] = {0};
int main(int argc, char** argv){
	int n;
	cin>>n;
	for(int i=0; i<n; i++){
		cin>>a[i];
	}
	int ans = 1e9;
	for(int low=0; low+17<=100; low++){
		int high = low + 17;
		int sum = 0;
		for(int i=0; i<n; i++){
			if(a[i] < low){
				sum += (low-a[i]) * (low-a[i]);
			}
			if(a[i]>high){
				sum += (high - a[i]) * (high - a[i]);
			}
		}
		if(sum < ans){
				ans = sum;
		}
	}
	cout << ans << endl;
	return 0;
}
```

02

子集枚举

例6-子数整数

逻辑航线

题目描述

对于一个五位数a1a2a3a4a5，可将其拆分为三个子数：
sub1=a1a2a3
sub2=a2a3a4
sub3=a3a4a5
例如，五位数20207可以拆分成
sub1=202
sub2=020(=20)
sub3=207
现在给定一个正整数K，要求你编程求出10000到30000之间所有满足下述条件的五位数，条件是这些五位数的三个子数sub1,sub2,sub3都可被K整除

输入格式

一个正整数K

输出格式

每一行为一个满足条件的五位数，要求从小到大输出。不得重复输出或遗漏。如果无解，则输出“No”。

题目地址： https://www.luogu.com.cn/problem/P1151

例6-解题思路

关键点在于如何对五位数N进行拆数?

前三位数 = N / 100
中间三位数 = N / 10 %1000
最后三位数 = N % 1000

例6-AC代码

逻辑航线

```cpp
#include <iostream>
#include <cstdio>
using namespace std;
int main(int argc, char** argv)
{
    int k;
    cin>>k;
    bool ok = false;
    for(int i=10000; i<=30000; i++){
        int a = i / 100;
        if(a % k == 0){
            int b = i / 10 % 1000;
            if( b % k == 0){
                int c = i % 1000;
                if(c % k == 0){
                    ok = true;
                    cout << i << endl;
                }
            }
        }
    }
    if(!ok){
        cout << "No";
    }
    return 0;
}
```

例7-最大乘积

逻辑航线

题目抽象

乘积最大的连续子集

题目地址： https://www.luogu.com.cn/problem/UVA11059

例7-解题思路

双重循环保证对每一个连续的子集进行计算

例7-AC代码

逻辑航线

```
#include<iostream>
#include<cstdio>
#include<cstring>
using namespace std;
int main(){
    long long n,m=0,maxValue;
    long long a[20] = {0};
    while (cin>>n){
        memset(a,0,sizeof(a));
        maxValue = 0;
        m++;
        for(int i=0; i<n; i++){
            cin>>a[i];
        }
        for(int i=0; i<n; i++){
            long long temp = a[i];
            if(temp > maxValue){
                maxValue = temp;
            }
            for(int j=i+1; j<n; j++){
                temp *= a[j];
                if(temp > maxValue){
                    maxValue = temp;
                }
            }
        }
        cout<<"Case #"<<m<<": The maximum product is
        "<<maxValue<<"."<<endl<<endl;
    }
    return 0;
}
```

例8-组合的输出

逻辑航线

题目描述

排列与组合是常用的数学方法，其中组合就是从n个元素中抽出rrr个元素(不分顺序且r≤n)，我们可以简单地将n个元素理解为自然数1,2,…,n从中任取r个数。

现要求你输出所有组合。

例如n=5,r=3所有组合为：

123,124,125,134,135,145,234,235,245,345

输入格式

一行两个自然数n,r(1<n<21,0≤r≤n)n,r(1<n<21,0 \le r \le n)n,r(1<n<21,0≤r≤n)。

输出格式

所有的组合，每一个组合占一行且其中的元素按由小到大的顺序排列，每个元素占三个字符的位置，所有的组合也按字典顺序。

**注意哦!输出时，每个数字需要3个场宽。

https://www.luogu.com.cn/problem/P1157

例8-解题思路

如何筛选子集

子集问题最常用的方法之一就是二进制法，我们先来回忆一下左移和右移。

给定数字N=3，我们将数字1左移N为后，其二进制形式就变成了1000，再将其减1，就变成了111，我们可以把111看成{1,2,3}这样的一个集合，同理101可以看成子集{1,3}。

根据这个性质，我们可以轻松化解大部分的子集类问题。

我们先将给定的N个数字的义为数字1左移N位的变量U，之后，再将U减1得到二进制数字S，S就是包含给定N个数字全体的二进制形式，即N个1的形式。然后对S遍历减1，并通过内置函数__bulitin_popcout统计当前的二进制数字包含1的个数是否与指定筛选的子集数目相等。

最后，我们再通过对U进行遍历右移1位，并与S进行与运算，确定到底存在哪一位数字，即可得出最终结果。

例8-AC代码

逻辑航线

```
#include <iostream>
#include <cstdio>
using namespace std;
int a[22] = {0};
int main(int argc, char** argv)
{
	int n,k,ans = 0;
	cin>>n>>k;
	for(int i=1 ; i<=n; i++)
	{
		a[i] = i;;
	}
	int u = 1<<n;
	for(int s=u-1; s>=0; s--)
	{
		if(__builtin_popcount(s) == k)
		{
			int sum=0;
			for(int j=0; j<=n; j++)
			{
				int temp = u>>j;
				if(s & temp)
				{
					printf("%3d",a[j]);
				}
			}
			cout<<endl;
		}
	}
	return 0;
}
```

例9-选数

逻辑航线

题目描述

已知 n 个整数 x1,x2,…, 以及1个整数k(k<n)。从n个整数中任选k个整数相加，可分别得到一系列的和。例如当n=4,k=3,4个整数分别为3,7,12,19时，可得全部的组合与它们的和为：

3+7+12=22
3+7+19=29
7+12+19=38
3+12+19=344

现在，要求你计算出和为素数共有多少种。
例如上例，只有一种的和为素数：3+7+19=29。

输入格式

键盘输入，格式为：n,k (1≤n≤20,k<n)
x1,x2,…,xn(1≤xi≤5000000)

输出格式

屏幕输出，格式为： 1个整数（满足条件的种数）。

题目地址：https://www.luogu.com.cn/problem/P1036

例9-解题思路

如何判断一个数字N是否为质数?
设置i=2开始遍历，直到i*i <= N。如果N % I == 0，则说明不是质数。

例9-AC代码


```cpp
#include <iostream>
#include <cstdio>

using namespace std;

int a[21] = {0};
bool check(int x)
{
    for(int i=2; i*i<x; i++)
    {
        if(x%i == 0)
        {
            return false;
        }
    }
    return true;
}
```

```cpp
int main(int argc, char** argv)
{
    int n,k,ans = 0;
    cin>>n>>k;
    for(int i=1 ; i<=n; i++)
    {
        scanf("%d",&a[i]);
    }
    int u = 1<<n;
    for(int s=u-1; s>=0; s--)
    {
        if(__builtin_popcount(s) == k)
        {
            int sum=0;
            for(int j=0; j<=n; j++)
            {
                int temp = u>>j;
                if(s & temp)
                {
                    sum += a[j];
                }
            }
            if(check(sum))
            {
                ans++;
            }
        }
    }
    printf("%d",ans);
    return 0;
}
```

03

排列枚举

例10-三连击（升级版）

逻辑航线

题目描述

将 1,2,…9共 9个数分成三组，分别组成三个三位数，且使这三个三位数的比例是 A:B:C，试求出所有满足条件的三个三位数，若无解，输出No!!!

输入格式

三个数，A、B、C。

输出格式

若干行，每行 3 个数字。按照每行第一个数字升序排列。

题目地址：https://www.luogu.com.cn/problem/P1618

例10-解题思路

题点，内置函数next_permulation的使用（STL），隶属algorithm头文件。
它能够按照字典序对一个数组进行排列，当不存在排列时，返回false。

函数结构：

```
Bool next_permutation (
    BidirectionalIterator_First //起始序列
    BidirectionalIterator_Last  //终止序列

);
```

例10-AC代码

逻辑航线

```
#include <iostream>
#include <cstdio>
#include <algorithm>
using namespace std;
typedef long long LL;
int a[10];
int main(int argc, char** argv){
    LL A,B,C,x,y,z,cnt = 0;
    cin >>A>>B>>C;
    for(int i=1; i<=9; i++){
        a[i] = i;
    }
    do{
        x = a[1] * 100 + a[2] * 10 + a[3];
        y = a[4] * 100 + a[5] * 10 + a[6];
        z = a[7] * 100 + a[8] * 10 + a[9];
        if(x*B == y*A && y*C == z*B){
                printf("%lld %lld %lld\n",x,y,z),cnt++;
        }
    }
    while(next_permutation(a+1,a+10));
    if(!cnt){
        cout<<"No!!!";
    }
    return 0;
}
```

例11-全排列

题目描述

输出自然数 1 到 n所有不重复的排列，即 n 的全排列，要求所产生的任一数字序列中不允许出现重复的数字。

输入格式

一个整数 n。

输出格式

由 1~ n组成的所有不重复的数字序列，每行一个序列。每个数字保留 5个场宽。

题目地址：https://www.luogu.com.cn/problem/P1706

例11-解题思路

逻辑航线

同例6。

例11-AC代码


```cpp
#include <iostream>
#include <cstdio>
#include <algorithm>
using namespace std;
int a[10] = {0};
int main(int argc, char** argv)
{
    int n;
    cin>>n;
    for(int i=1; i<=n; i++){
        a[i] = i;
    }
    do{
        for(int i=1; i<=n; i++){
                printf("%5d",a[i]);
        }
        cout<<endl;
    }
    while(next_permutation(a+1,a+n+1));
    return 0;
}
```

例12-火星人

题目抽象：

对于n的全排列，找到一个给定排列向后的第m个排列。

题目地址：https://www.luogu.com.cn/problem/P1088

例12-AC代码

逻辑航线

```
#include <iostream>
#include <cstdio>
#include <algorithm>
using namespace std;
int a[10001] = {0};
int main(int argc, char** argv){
    int n,m;
    cin>>n>>m;
    for(int i=1; i<=n; i++){
        cin>>a[i];
    }
    for(; m--;){
        next_permutation(a+1,a+n+1);
    }
    for(int i=1; i<=n; i++){
    C   out<<a[i]<<" ";
    }
    return 0;
}
```

04

课后练习

练习1-涂国旗

逻辑航线

题目描述

某国法律规定，只要一个由 N×M个小方块组成的旗帜符合如下规则，就是合法的国旗。（毛熊：阿嚏——）

从最上方若干行（至少一行）的格子全部是白色的；
接下来若干行（至少一行）的格子全部是蓝色的；
剩下的行（至少一行）全部是红色的；

现有一个棋盘状的布，分成了 N行 M列的格子，每个格子是白色蓝色红色之一，小a 希望把这个布改成该国国旗，方法是在一些格子上涂颜料，盖住之前的颜色。

小a很懒，希望涂最少的格子，使这块布成为一个合法的国旗。

输入格式

第一行是两个整数 N,M。

接下来 N 行是一个矩阵，矩阵的每一个小方块是W（白），B（蓝），R（红）中的一个。

输出格式

一个整数，表示至少需要涂多少块。

题目地址：https://www.luogu.com.cn/problem/P3392

练习1-AC代码

逻辑航线

```
#include <cstdio>
int n,m,ans=1e9;
char a[52][52];
int main()
{
	scanf("%d%d",&n,&m);
	for (int i=0; i<n; i++)
		scanf("%s",a[i]);
	for (int i=0; i<n-2; i++)
		for (int j=i+1; j<n-1; j++)
		{
			int tot=0;
			for (int x=0; x<=i; x++)
				for (int y=0; y<m; y++)
					if (a[x][y]!='W')
						tot++; //染白需几步
			for (int x=i+1; x<=j; x++)
				for (int y=0; y<m; y++)
					if (a[x][y]!='B')
						tot++; //染蓝需几步
			for (int x=j+1; x<n; x++)
				for (int y=0; y<m; y++)
					if (a[x][y]!='R')
						tot++; //染红需几步
			if (tot<ans)
				ans=tot;
		}
	printf("%d",ans);
	return 0;
}
```

练习2-First Step

逻辑航线

题目抽象：

由N * M小方格组成的篮球场，每个小方格可能是空地”.”也可能是障碍”#”，求站位的方案数。

题目地址：https://www.luogu.com.cn/problem/P3654

练习2-AC代码

逻辑航线

```
#include<iostream>
#include<cstdio>
#include<algorithm>
using namespace std;
int r, c, k;
char pos[110][110];
int main(){
     scanf("%d%d%d", &r, &c, &k);
     int ans = 0;
     for (int i = 0; i < r; i++){
          int tmp = 0;
          for (int j = 0; j < c; j++){
               cin >> pos[i][j];
               if (pos[i][j] == '.')
                    tmp++;
               else if (tmp > 0 && pos[i][j] == '#'){
                    if (tmp >= k)
                    ans += (tmp - k + 1);
                    tmp = 0;
               }
          }
          if (tmp >= k)
                    ans += (tmp - k + 1);
     }
     for (int j = 0; j < c; j++){
          int tmp = 0;
          for (int i = 0; i < r; i++){
               if (pos[i][j] == '.')
                    tmp++;
               else if (tmp > 0 && pos[i][j] == '#'){
                    if (tmp >= k)
                    ans += (tmp - k + 1);
                    tmp = 0;
               }
          }
          if (tmp >= k)
                    ans += (tmp - k + 1);
     }
     if (k == 1)
          ans /= 2;
     printf("%d\n", ans);
     return 0;
}
```

练习3-回文质数

逻辑航线

题目描述

因为 151 既是一个质数又是一个回文数（从左到右和从右到左是看一样的），所以 151 是回文质数。

写一个程序来找出范围 [a,b](5≤a<b≤100,000,000)[a,b] (5 \le a < b \le 100,000,000)[a,b](5≤a<b≤100,000,000)(一亿)间的所有回文质数。

输入格式

第 1 行: 二个整数 a 和 b .

输出格式

输出一个回文质数的列表，一行一个。

题目地址：https://www.luogu.com.cn/problem/P1217

练习3-解题思路

逻辑航线

关键结论：

1、偶数位回文数（除了11）必定不是质数。
2、偶数肯定不是质数。

根据关键结论1，我们在本题中需要求解5-100 000 000中的回文质数，因为100 000 000肯定不是质数，再加上10 000 000 - 99 999 999这些都是偶数位的数，他们中的回文数肯定不是质数，所以也可以直接排除。

因此，我们只需要枚举5 - 10 000 000之间的数字即可。

关键步骤：

找出所有的回文数，再判断他们是不是质数

练习3-AC代码

逻辑航线

```c
#include<stdio.h>
#include<math.h>
int IsPrime(int n){
    if(n <= 1)
        return 0;
    if(n == 2)
        return 1;
    for(int i = 2; i <= (int)sqrt(n); i++)
        if(n%i==0)
            return 0;
        return 1;
}
int IsPalindrome(int n){
    int temp = n;
    int sum = 0;
    while (n != 0){
        sum = sum * 10 + n % 10;
        n /= 10;
    }
    if (sum == temp)
        return 1;
    else
        return 0;
}
```

```c
int main(){
    int begin, end;
    scanf("%d%d",&begin,&end);
    if(begin%2==0)
        begin += 1;

    if(end>9999999)
        end = 9999999;


    for (int i = begin; i <= end; i += 2){
        if (IsPalindrome(i)&&IsPrime(i))
            printf("%d\n",i);
    }
    return 0;
}
```

作业4-火柴棒等式

逻辑航线

题目地址：https://www.luogu.com.cn/problem/P1149

作业4-解题思路

逻辑航线

因为输入n小于等于24，所以我们能够推测出，当前火柴棒能够组成的最大值等式为：1111 + 0 = 1111，共消耗了26根火柴，只需在0~1111内枚举两个数字，使得该等式组成的火柴棒个数为n。

作业4-AC代码

逻辑航线

```
#include<iostream>
#include<cstdio>
using namespace std;
int b[10]= {6,2,5,5,4,5,6,3,7,6};
int a[2250];
int main()
{
    int n,ans;
    scanf("%d",&n);
    n-=4;
    ans=0;
    for(int i=0; i<=9; i++)
        a[i]=b[i];
    for(int i=10; i<=2250; i++)
        a[i]=a[i/10]+a[i%10];
    for(int i=0; i<=1111; i++)
        for(int j=0; j<=1111; j++)
            if(a[i]+a[j]+a[i+j]==n)ans++;
    printf("%d\n",ans);
}
```

感　谢　观　看

联系地址：河北省廊坊市大厂县孔雀英国宫二期