练习2

#include<iostream>

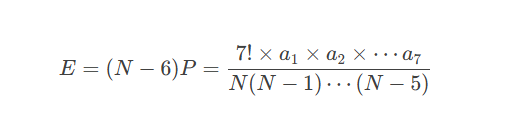
using namespace std;

/\*

全是数学！计算期望，高中应该没问题

看公式吧，题解里有详细思路

[题解 P3802 【小魔女帕琪】 - 洛谷专栏](https://www.luogu.com.cn/article/a5wystwm)



\*/

int main() {

double a[8], sum = 0,result;

for (int i = 1; i < 8; ++i) {

cin >> a[i];

sum += a[i];

}

result = a[1] / sum \* 1.0 \* a[2] / (sum - 1) \* a[3] / (sum - 2) \* a[4] / (sum - 3) \* a[5] / (sum - 4) \* a[6] / (sum - 5) \* a[7] \* 5040;

printf("%.3f", result);

return 0;

}

练习3

#include<iostream>

#include<vector>

using namespace std;

/\*

\* 基本用的都是题解思路：

使用到了第二类什么斯特林，题解里有讲到这个玩意，有兴趣可以去看看，我半理解半应用

https://zhuanlan.zhihu.com/p/350774728

思路：计算n的质因数，统计个数

然后用到第二类斯特林，也就是计算排列组合 从统计的个数中选k个数的方案数，C 下标count1 上标k

但是如果暴力计算，超时，所以使用他这个第二类斯特林计算方法，快速求得各种排列组合答案

该题要理解质因数的种类作用（只需要种类数就可以计算方案数），以及排列组合的计算方法

\*/

long long t, n, k;//最多八个质数

int s[9][9];

void jisuan1() {

s[0][0] = 1;

// 第二类Stirling数计算，最后输出s[count1][k]， 索引含义是什么

for (int i = 1; i <= 8; ++i) {

for (int j = 1; j <= 8; ++j) {

s[i][j] = j \* s[i - 1][j] + s[i - 1][j - 1];

}

}

}

int jisuan(long long n, long long k) {

int count1 = 0;

int n1 = n;

for (int i = 2; i \* i <= n; ++i) {

if (n % i == 0) {

count1++;

}

while (n % i == 0) {

n /= i;//避免出现i的倍数导致错乱

}

}

if (n > 1) {

count1++;

} // 最后分解出来的数

if (count1 < k) {//数量不满足要求

return 0;

}

else {

//int a, b; 超时

//for (int i = count1; i > k; --i) {

// a \*= i;

//}

//for (int i = k; i > 0; ++i) {

// b \*= i;

//}

//return a / b;

return s[count1][k];

}

}

int main() {

cin >> t;

jisuan1();

while (t--) {

cin >> n >> k;

cout << jisuan(n, k) << endl;

}

return 0;

}

练习1

//神经网络的东西，训练数据，计算一个平衡点，我不会搞，用py的比较好

//可以看题解思路

// [题解 P3691 【[PUB1]妖精大战争】 - 洛谷专栏](https://www.luogu.com.cn/article/25b7nlpv)

#**include**<iostream>

#**include**<cstdio>

#**include**<cstring>

using namespace std;

int n,m,k;

int sunny[233][233];

int lunar[233][233];

double len=1.0;

double x,y;int t;

int main()

{

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

if(n<=1000)len=8.0;

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

{

scanf("%lf%lf%d",&x,&y,&t);

if(t==1)sunny[(int)(x/len)][(int)(y/len)]++;

else lunar[(int)(x/len)][(int)(y/len)]++;

}

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

{

scanf("%lf%lf",&x,&y);

if(sunny[(int)(x/len)][(int)(y/len)]>=lunar[(int)(x/len)][(int)(y/len)])

printf("1\n");

else printf("-1\n");

}

return 0;

}