// pair 用法

pair<int,int>p[MAXN];

p[i]=make\_pair(x,y);

dis(p[i],p[j]);

// multiset<char> st;//可重复的以字典序的方式存字符

multiset<char>::iterator it;

map<LL,int>::iterator it=dp.lower\_bound(-sum);

// 结构体 排序

double x, y;

bool operator < (const Node b) const {

return x < b.x;

}

double dis(pair<int,int>p1,pair<int,int>p2)

{

return sqrt((double)(p1.first-p2.first)\*(p1.first-p2.first)+(p2.second-p1.second)\*(p2.second-p1.second));

}

// 组合数

int c()

{

C[1][0] = C[1][1] = 1;

for (int i = 1; i <= 71; i++)

C[i][0] = 1;

for (ll i = 2; i <= 71; i++)

for (ll j = 1; j <= i; j++)

{

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

}

}

// 或者

LL C(LL n, LL k) {

if(n - k < k) k = n - k;

LL ans = 1;

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

ans = ans \* (n - i + 1) / i;

return ans;

}