#include <bits/stdc++.h>

using namespace std;

vector<int>r1[3];

#define MAX 100000000;

struct ansVal {

int r[3];

int minVal;

};

ansVal solve(int a, int b, int c){

ansVal answer;

answer.minVal = MAX;

for(int i=0;i<r1[b].size(); i++){

int leftVal = upper\_bound(r1[a].begin(), r1[a].end(), r1[b][i]) - r1[a].begin();

if (leftVal == 0) {

continue;

}

leftVal--;

int rightVal = lower\_bound(r1[c].begin(), r1[c].end(), r1[b][i]) - r1[c].begin();

if(rightVal == r1[c].size()) {

continue;

}

leftVal = r1[a][leftVal];

rightVal = r1[c][rightVal];

int val = ((leftVal - r1[b][i])\*(leftVal - r1[b][i])) + ((rightVal - r1[b][i])\*(rightVal - r1[b][i])) +

((rightVal - leftVal)\*(rightVal - leftVal));

if(val < answer.minVal ) {

answer.minVal = val;

answer.r[b] = r1[b][i];

answer.r[a] = leftVal;

answer.r[c] = rightVal;

}

}

return answer;

}

int main() {

vector<int>v = {2, 7, 7, 11, 11, 13, 14, 14, 14, 17};

int l = 7, r = 14;

//int ans = upper\_bound(v.begin(), v.end(), r) - lower\_bound(v.begin(), v.end(), l);

//cout<<ans<<endl;

vector<int>v1 = {7, 2, 1, 16, 100, 10000, 1, 1, 2, 3, 13};

// Compression

vector<int>v2 = v1;

sort(v2.begin(), v2.end());

for(int i=0;i<v1.size();i++){

v1[i] = lower\_bound(v2.begin(), v2.end(), v1[i]) - v2.begin();

}

// Mapping

/\* for(int i=0;i<v1.size();i++){

cout<<v1[i]<<" ";

}

cout<<endl;\*/

int prefix[v1.size()], suffix[v1.size()];

memset(prefix, 0, sizeof(prefix));

memset(suffix, 0, sizeof(suffix));

for(int i=2;i<v1.size();i++){

suffix[v1[i]]++;

}

prefix[v1[0]]++;

int ans = 0;

for(int i=1;i<v1.size();i++){

int val = v1[i];

int leftAns = 0, rightAns = 0;

for(int j=val;j>=0;j--){

leftAns += prefix[j];

}

for(int j=val;j<v1.size();j++){

rightAns += suffix[j];

}

ans += leftAns \* rightAns;

}

cout<<ans<<endl;

// Candies

int n = 4;

for(int i=0;i<4;i++){

r1[0].push\_back(i\*10 + 10);

r1[1].push\_back(3\*10 + i);

r1[2].push\_back(2\*i\*i + 13);

}

for(int i=0;i<3;i++){

sort(r1[i].begin(), r1[i].end());

}

// 0 - red, 1 - blue, 2-green

vector<ansVal>a1(6);

a1[0] = solve(0, 1, 2);

a1[1] = solve(1, 2, 0);

a1[2] = solve(0, 2, 1);

a1[3] = solve(2, 1, 0);

a1[4] = solve(2, 0, 1);

a1[5] = solve(1, 0, 2);

ansVal answer;

answer.minVal = MAX;

for(int i=0; i<6;i++){

if(a1[i].minVal < answer.minVal){

answer = a1[i];

}

}

cout<<answer.r[0] <<" "<<answer.r[1]<<" "<< answer.r[2]<<endl;

}