牛客2025秋季算法编程训练联赛2-基础组

A.做游戏

Code

C++风格:

```cpp
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define double long double
#define endl '\n'
const int mod = 1e9 + 7;
const int N = 3e5 + 7;

void solve()
{
    int a,b,c;
    cin >> a >> b >> c;
    int x, y, z;
    cin >> x >> y >> z;

    cout << min(a, y) + min(b, z) + min(c, x) << endl;
    return ;
}

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int T = 1;
    // cin >> T;
    while (T--) {
        solve();
    }
    return 0;
}
```

C语言风格:

```c
#include <stdio.h>
#include <stdlib.h>

#define min(a, b) ((a) < (b) ? (a) : (b))

void solve()
{
    long long a, b, c;
    scanf("%lld %lld %lld", &a, &b, &c);
    long long x, y, z;
    scanf("%lld %lld %lld", &x, &y, &z);

    printf("%lld\n", min(a, y) + min(b, z) + min(c, x));
    return ;
}

int main()
{
    int T = 1;
    // scanf("%d", &T);
    while (T--) {
        solve();
    }
    return 0;
}
```

B.排数字

C++风格:

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define double long double
#define endl '\n'
const int mod = 1e9 + 7;
const int N = 3e5 + 7;

void solve()
{
    int cnt[10] = {};

    int n;
    string s;

    cin >> n >> s;

    for(char x : s)
    {
        cnt[x - '0']++;
    }

    cout << min(cnt[1], cnt[6] - 1) << endl;
    return ;
}

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int T = 1;
    // cin >> T;
    while (T--) {
        solve();
    }
    return 0;
}
```

C语言风格:

```
#include <stdio.h>
#include <string.h>

#define min(a, b) ((a) < (b) ? (a) : (b))

void solve()
{
    long long cnt[10] = {0};  // 初始化计数数组

    long long n;
    char s[300007];  // 适配原代码中N=3e5+7的长度

    scanf("%lld %s", &n, s);  // 读取输入

    // 统计每个数字的出现次数
    for (int i = 0; s[i] != '\0'; i++)
    {
        char x = s[i];
        cnt[x - '0']++;
    }

    // 计算并输出结果
    printf("%lld\n", min(cnt[1], cnt[6] - 1));
    return ;
}

int main()
{
    int T = 1;
    // scanf("%d", &T);  // 如需多组测试数据，取消注释
    while (T--)
    {
        solve();
    }
    return 0;
}
```

C.判正误

C++风格:

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define double long double
```

```
#define endl '\n'
const int N = 3e5 + 7;

int binpow(int a, int b, int p)
{
    int res = 1;
    while (b > 0)
    {
        if (b & 1)
        {
            res = res * a % p;
        }
        a = a * a % p;
        b >>= 1;
    }
    return res;
}

void solve()
{
    int a, b, c, d, e, f, g;
    cin >> a >> b >> c >> d >> e >> f >> g;

    auto fun = [&](int a, int b, int c, int d, int e, int f, int g, int mod) ->
bool
    {
        int sum = 0;
        int f1 = 1, f2 = 1, f3 = 1;
        if (a < 0)
        {
            a = abs(a);
            f1 = -1;
        }
        if (b < 0)
        {
            b = abs(b);
            f2 = -1;
        }
        if (c < 0)
        {
            c = abs(c);
            f3 = -1;
        }

        sum = (sum + f1 * binpow(a, d, mod) + mod) % mod;
        sum = (sum + f2 * binpow(b, e, mod) + mod) % mod;
        sum = (sum + f3 * binpow(c, f, mod) + mod) % mod;
        g = (g + mod) % mod;
        return sum == g;
    };

    bool ans = true;
    ans = ans & fun(a, b, c, d, e, f, g, 1e9 + 7);
    ans = ans & fun(a, b, c, d, e, f, g, 1e9 + 6);

    cout << (ans ? "Yes" : "No") << endl;
    return;
}

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int T = 1;
    cin >> T;
    while (T--)
    {
        solve();
    }
    return 0;
}
```

C语言风格:

```
#include <stdio.h>
#include <stdlib.h>  // 用于llabs函数

#define ll long long  // 替代原代码的int long long
const int N = 300007;  // 原代码中的N=3e5+7

// 快速幂函数，计算(a^b) mod p
ll binpow(ll a, ll b, ll p) {
    ll res = 1;
    while (b > 0) {
        if (b & 1) {
            res = res * a % p;
        }
        a = a * a % p;
        b >>= 1;
    }
    return res;
}

// 替代原lambda表达式fun的功能函数
int fun(ll a, ll b, ll c, ll d, ll e, ll f, ll g, ll mod) {
    ll sum = 0;
    ll f1 = 1, f2 = 1, f3 = 1;

    // 处理负数情况，记录符号并取绝对值
    if (a < 0) {
        a = llabs(a);  // long long类型的绝对值
        f1 = -1;
    }
    if (b < 0) {
        b = llabs(b);
        f2 = -1;
    }
    if (c < 0) {
        c = llabs(c);
        f3 = -1;
    }

    // 计算各项并累加，确保结果非负后取模
    sum = (sum + f1 * binpow(a, d, mod) + mod) % mod;
    sum = (sum + f2 * binpow(b, e, mod) + mod) % mod;
    sum = (sum + f3 * binpow(c, f, mod) + mod) % mod;
    g = (g + mod) % mod;  // 处理g为负数的情况

    return (sum == g) ? 1 : 0;  // 1表示true，0表示false
}

void solve() {
    ll a, b, c, d, e, f, g;
    // 读取7个long long类型变量
    scanf("%lld %lld %lld %lld %lld %lld %lld", &a, &b, &c, &d, &e, &f, &g);

    int ans = 1;  // 初始为true
    // 检查两个模的情况，用逻辑与连接结果
    ans &= fun(a, b, c, d, e, f, g, 1000000007LL);  // 1e9+7
    ans &= fun(a, b, c, d, e, f, g, 1000000006LL);  // 1e9+6

    // 输出结果
    printf("%s\n", ans ? "Yes" : "No");
}

int main() {
    int T;
    scanf("%d", &T);  // 读取测试用例数量
    while (T--) {
        solve();
    }
    return 0;
}
```

D.数三角

C++风格:

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define double long double
#define endl '\n'

void solve()
{
    int n;
    cin >> n;

    vector<pair<int, int>> p(n);
    for (int i = 0; i < n; ++i)
    {
        cin >> p[i].first >> p[i].second;
    }

    int ans = 0;
    for (int i = 0; i < n; ++i)
    {
        for (int j = i + 1; j < n; ++j)
        {
            for (int k = j + 1; k < n; ++k)
            {
                int x1 = p[j].first - p[i].first;
                int y1 = p[j].second - p[i].second;
                int x2 = p[k].first - p[i].first;
                int y2 = p[k].second - p[i].second;
                int hh = x1 * y2 - y1 * x2;
                if (hh == 0)
                {
                    continue;
                }

                int d1 = (p[i].first - p[j].first) * (p[i].first - p[j].first) + (p[i].second - p[j].second) * (p[i].second - p[j].second);
                int d2 = (p[i].first - p[k].first) * (p[i].first - p[k].first) + (p[i].second - p[k].second) * (p[i].second - p[k].second);
                int d3 = (p[j].first - p[k].first) * (p[j].first - p[k].first) + (p[j].second - p[k].second) * (p[j].second - p[k].second);
                int mx = max(d1, max(d2, d3));
                int sum = d1 + d2 + d3;
                if (mx > sum - mx)
                {
                    ans++;
                }
            }
        }
    }

    cout << ans << '\n';
}

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int T = 1;
    // cin >> T;
    while (T--)
    {
        solve();
    }
    return 0;
}
```

C语言风格:

```
#include <stdio.h>

long long max (long long x, long long y, long long z) {
    if (x >= y && x >= z) {
        return x;
    }else if (y >= x && y >= z) {
        return y;
    }else {
        return z;
    }
    return 0;
}
int main () {
    int n = 0;
    scanf ("%d", &n);
    int arr[505][2];
    for (int s = 0; s < n; s++) {
        scanf ("%d%d", &arr[s][0], &arr[s][1]);
    }
    long long p = 0;
    for (int a = 0; a < n; a++) {
        for (int b = a + 1; b < n; b++) {
            for (int c = b + 1; c < n; c++) {
                int x1 = arr[a][0], y1 = arr[a][1];
                int x2 = arr[b][0], y2 = arr[b][1];
                int x3 = arr[c][0], y3 = arr[c][1];
                long long cross = (long long)(x2 - x1) * (y3 - y1) - (long long)
(y2 - y1) * (x3 - x1);
                if (cross == 0) {
                    continue;
                }
                long long x = (long long) ((arr[a][0] - arr[b][0]) *( arr[a][0] -
arr[b][0])) + (long long) ((arr[a][1] - arr[b][1]) *(arr[a][1] - arr[b][1]));
                long long y = (long long) ((arr[a][0] - arr[c][0]) *( arr[a][0] -
arr[c][0])) + (long long) ((arr[a][1] - arr[c][1]) *(arr[a][1] - arr[c][1]));
                long long z = (long long) ((arr[c][0] - arr[b][0]) *( arr[c][0] -
arr[b][0])) + (long long) ((arr[c][1] - arr[b][1]) *(arr[c][1] - arr[b][1]));
                long long ma = max(x, y, z);
                if (ma == x && y + z < x) {
                    p++;
                }else if (ma == y && z + x < y) {
                    p++;
                }else if (ma == z && x + y < z) {
                    p++;
                }
            }
        }
    }
    printf ("%lld", p);
    return 0;
}
```

E.做计数

C++风格:

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define double long double
#define endl '\n'
const int mod = 1e9 + 7;
const int N = 3e5 + 7;

void solve()
{
    int n;
    cin >> n;

    int ans = 0;
    for (int i = 1; i * i <= n; i++)
    {
        int z = i * i;
        for (int j = 1; j * j <= z; j++)
        {
            if(z % j == 0)
            {
                ans++;
                if(z / j != i)
                {
                    ans++;
                }
            }
        }
    }
    cout << ans << endl;

    return;
}
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int T = 1;
    // cin >> T;
    while (T--)
    {
        solve();
    }
    return 0;
}
```

C语言风格:

```c
#include <stdio.h>
```

```
#include <math.h>  // 用于循环中的平方判断（隐含sqrt逻辑）

#define mod 1000000007LL  // 原代码中的mod常量
#define N 300007LL        // 原代码中的N常量

typedef long long ll;  // 替代原代码的#define int long long

void solve() {
    ll n;
    scanf("%lld", &n);  // 读取输入n

    ll ans = 0;
    // 外层循环：i从1到sqrt(n)
    for (ll i = 1; i * i <= n; i++) {
        ll z = i * i;  // 计算z = i²
        // 内层循环：j从1到sqrt(z)
        for (ll j = 1; j * j <= z; j++) {
            if (z % j == 0) {  // 若j是z的约数
                ans++;
                if (z / j != i) {  // 若另一个约数不等于i，额外计数
                    ans++;
                }
            }
        }
    }
    printf("%lld\n", ans);  // 输出结果
}

int main() {
    int T = 1;
    // scanf("%d", &T);  // 如需多组测试数据，取消注释
    while (T--) {
        solve();
    }
    return 0;
}
```

F.拿物品

C++风格:

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define double long double
#define endl '\n'
const int mod = 1e9 + 7;
const int MAX_N = 2005;

void solve()
{
    int n;
    cin >> n;
    vector<int> a(n), b(n);
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
    for (int i = 0; i < n; i++)
    {
        cin >> b[i];
    }

    vector<tuple<int, int, int>> ob;
    for (int i = 0; i < n; i++)
    {
        ob.emplace_back(a[i], b[i], i + 1);
    }

    sort(ob.begin(), ob.end(), [](const auto &x, const auto &y)
         {
        int sum_x = get<0>(x) + get<1>(x);
        int sum_y = get<0>(y) + get<1>(y);
        return sum_x > sum_y; });

    vector<int> ansA;
    vector<int> ansB;

    for (int i = 0; i < n; i++)
    {
        if (i % 2 == 0)
        {
            ansA.push_back(get<2>(ob[i]));
        }
        else
        {
            ansB.push_back(get<2>(ob[i]));
        }
    }

    for (int x : ansA)
    {
        cout << x << ' ';
    }
    cout << endl;
    for (int x : ansB)
    {
        cout << x << ' ';
    }
    cout << endl;
}

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int T = 1;
    // cin >> T;
    while (T--)
    {
        solve();
    }
    return 0;
}
```
