

# 0x90 杂

---

## 0x91 输入输出语法

- 输出左/右对齐：
  - `cout << setw(x) << n`, 默认右对齐, 并设置字段宽度为 `x`。
  - `cout << left << setw(x) << n`, 左对齐, 并设置字段宽度为 `x`。
- 浮点数位
  - `cout << fixed << setprecision(x)`, 输出小数点后 `x` 位。

## 0x92 高精度

- 高精度 + 高精度

```
#include<bits/stdc++.h>

using namespace std;

#define int long long

const int N = 1e5 + 10, inf = 1e18;

typedef pair<int, int> PII;

void trans(string a, vector<int> &A)
{
    for (int i = a.size() - 1; i >= 0; i -- ) A.push_back(a[i] - '0');
}

bool cmp(string a, string b)
{
    if (a.size() != b.size()) return a.size() > b.size();
    for (int i = 0; i < a.size(); i ++ )
        if (a[i] != b[i])
            return a[i] > b[i];
    return true;
}

string sub(string a, string b);

string add(string a, string b) // 高精度加法
{
    if (a[0] == '-' && b[0] == '-') return '-' + add(a.substr(1), b.substr(1));
    else if (a[0] == '-') return sub(b, a.substr(1));
    else if (b[0] == '-') return sub(a, b.substr(1));

    if (a.size() < b.size()) return add(b, a);
```

```
vector<int> A, B;
trans(a, A), trans(b, B);

vector<int> C;
int t = 0;
for (int i = 0; i < A.size(); i++)
{
    t += A[i];
    if (i < B.size()) t += B[i];
    C.push_back(t % 10);
    t /= 10;
}
if (t) C.push_back(t);

string c = "";
for (int i = C.size() - 1; i >= 0; i--) c += (char)C[i] + '0';
return c;
}

string sub(string a, string b) // 高精度减法
{
    if (a[0] == '-' && b[0] == '-') return sub(b.substr(1), a.substr(1));
    else if (a[0] == '-') return '-' + add(a.substr(1), b);
    else if (b[0] == '-') return add(a, b.substr(1));

    if (!cmp(a, b)) return '-' + sub(b, a);

    vector<int> A, B;
    trans(a, A), trans(b, B);

    vector<int> C;
    for (int i = 0, t = 0; i < A.size(); i++)
    {
        t = A[i] - t;
        if (i < B.size()) t -= B[i];
        C.push_back((t + 10) % 10);
        if (t < 0) t = 1;
        else t = 0;
    }
    while (C.size() > 1 && C.back() == 0) C.pop_back();

    string s = "";
    for (int i = C.size() - 1; i >= 0; i--) s += (char)C[i] + '0';
    return s;
}

string mul(string a, string b) // 高精乘高精
{
    if (a[0] == '-' && b[0] == '-') return mul(a.substr(1), b.substr(1));
    else if (a[0] == '-') return '-' + mul(a.substr(1), b);
    else if (b[0] == '-') return '-' + mul(a, b.substr(1));

    vector<int> A, B;
```

```
trans(a, A), trans(b, B);

int num = a.size() + b.size() - 1;
vector<int> C(num + 1);
for (int i = 0; i < num; i ++ )
{
    for (int j = 0; j <= i; j ++ )
        C[i] += (j < a.size() ? A[j] : 0) * ((i - j) < b.size() ? B[i - j] :
0);

    if (C[i] >= 0)
    {
        C[i + 1] += C[i] / 10;
        C[i] %= 10;
    }
}
while (C.size() > 1 && C.back() == 0) C.pop_back();

string s = "";
for (int i = C.size() - 1; i >= 0; i -- ) s += (char)C[i] + '0';
return s;
}

string mul(string a, int b) // 高精度乘法
{
    if (a[0] == '-' && b < 0) return mul(a.substr(1), -b);
    else if (a[0] == '-') return '-' + mul(a.substr(1), b);
    else if (b < 0) return '-' + mul(a, -b);

    vector<int> A;
    trans(a, A);

    vector<int> C;
    int t = 0;
    for(int i = 0; i < A.size() || t; i ++ )
    {
        if(i < A.size()) t += A[i] * b;
        C.push_back(t % 10);
        t /= 10;
    }
    while(C.size() > 1 && C.back() == 0) C.pop_back();

    string s = "";
    for (int i = C.size() - 1; i >= 0; i -- )
    {
        s += C[i] % 10 + '0';
        C[i] /= 10;
        while (C[i])
        {
            s += (char)(C[i] % 10) + '0';
            C[i] /= 10;
        }
    }
    return s;
}
```

```

}

string div(string a, int b, int &r) // 高精度除法带余数
{
    vector<int> A, C;
    trans(a, A);
    r = 0;
    for(int i = A.size() - 1; i >= 0; i -- )
    {
        r = r * 10 + A[i];
        C.push_back(r / b);
        r %= b;
    }
    reverse(C.begin(), C.end());
    while(C.size() > 1 && C.back() == 0) C.pop_back();

    string s = "";
    for (int i = C.size() - 1; i >= 0; i -- ) s += (char)C[i] + '0';
    return s;
}

signed main()
{
    ios::sync_with_stdio(0); cin.tie(0); cin.tie(0);

    string a, b;
    cin >> a >> b;

    cout << add(a, b) << '\n';

    return 0;
}

```

## 0x93 结构体中重载运算符

```

bool operator> (const Node& W) const
{
    return w > W.w;
}

```

## 0x94 对拍

- 首先需要创建一个数据生成器 *data.cpp*, 并且编译生成 *data.exe* 文件。

```

#include <bits/stdc++.h>

using namespace std;

```

```
int main()
{
    ios::sync_with_stdio(0); cin.tie(0); cin.tie(0);

    srand(time(0));
    rand();
    cout << rand() % 5 << "\n";

    return 0;
}
```

- 再创建一个对拍文件

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    ios::sync_with_stdio(0); cin.tie(0); cin.tie(0);

    while (1)
    {
        system("data.exe > in.txt");
        system("WA.exe < in.txt > out.txt");
        system("AC.exe < in.txt > ans.txt");
        if (system("fc out.txt ans.txt")) return 1;
    }

    return 0;
}
```