2025

L1-1 珍惜生命（“Hello World!”）

```
print("Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.")
```

L2-2 偷感好重（简单运算）

```
#include <bits/stdc++.h>
using namespace std;

int main() {

    int a, b, c;
    cin >> a >> b >> c;
    cout << a + b + c;

    return 0;
}
```

L1-3 高温补贴（if - else）

```
#include <bits/stdc++.h>
using namespace std;

int main() {

    int T, s, t;
    cin >> T >> s >> t;

    if(T >= 35 && s == 1 && t >= 33) {
        cout << "Bu Tie\n";
        cout << T << '\n';
    } else if(s == 0 && T >=35 && t >= 33) {
        cout << "Shi Nei\n";
        cout << T << '\n';
    } else if(s == 1) {
        cout << "Bu Re\n";
        cout << t << '\n';
    } else {
        cout << "Shu Shi\n";
        cout << t << '\n';
    }

    return 0;
}
```

L1-4 零头就抹了吧（简单while）

```
#include <bits/stdc++.h>
using namespace std;
```

```cpp

int main() {

    int n;
    cin >> n;

    int i = 1;
    while(i <= n) {
        i *= 2;
    }
    cout << i / 2;

    return 0;
}
```

L1-5 这是字符串题（简单数组计数）

```cpp
#include <bits/stdc++.h>
using namespace std;

string s;
int f[26], cnt[26];

int main() {


    cin >> s;
    for(int i = 0; i < 26; ++i) cin >> f[i];

    int sum = 0;
    for(size_t i = 0; i < s.size(); ++i) {
        sum += f[s[i] - 'a'];
        cnt[s[i] - 'a']++;
    }

    for(int i = 0; i < 26; ++i) {
        if(i) cout << ' ';
        cout << cnt[i];
    }
    cout << '\n';
    cout << sum;

    return 0;
}
```

L1-6 这不是字符串题（典型字符串EX题）

挺麻烦的QAQ

1. 数组序列转字符串 `s1 += (char)(x - 1 + 'a');`
2. 查找 + 替换

   `if(s1.find(b1) != string::npos)`

   `{s1.replace(s1.find(b1), a1, b2);}`

3. 求值 + 插入

   `char ch = (char)(val / 2 + 'a');`

   `s1.insert(i, 1, ch);`   insert(索引，长度，字符串)

4. 区间反转

   `reverse(s1.begin() + (l - 1), s1.begin() + r);`

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {

    int n, m;
    cin >> n >> m;
    string s1 = "";
    for(int i = 0; i < n; ++i) {
        int x;
        cin >> x;
        s1 += (char)(x - 1 + 'a');
    }

    while(m--) {

        int op;
        cin >> op;

        if(op == 1) {

            int a1, a2;
            string b1 = "", b2 = "";

            cin >> a1;
            for(int i = 0; i < a1; ++i) {
                int x;
                cin >> x;
                b1 += (char)(x - 1 + 'a');
            }

            cin >> a2;
            for(int i = 0; i < a2; ++i) {
                int x;
                cin >> x;
                b2 += (char)(x - 1 + 'a');
            }

            if(s1.find(b1) != string::npos) {
                s1.replace(s1.find(b1), a1, b2);
            }

        } else if(op == 2) {

            for(size_t i = 1; i < s1.size(); ++i) {
                int val = (s1[i - 1] - 'a') + (s1[i] - 'a');
                if(val % 2 == 0) {
                    char ch = (char)(val / 2 + 'a');
                    s1.insert(i, 1, ch);
                    i++;
                }
            }

        } else if(op == 3) {
            int l, r;
            cin >> l >> r;
            reverse(s1.begin() + (l - 1), s1.begin() + r);
        }

    }

    for(size_t i = 0; i < s1.size(); ++i) {
        if(i != 0) cout << ' ';
        cout << (int)(s1[i] - 'a' + 1);
    }

    return 0;
}
```

L1-7 大幂数（暴力）

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

int main() {

    ll n;
    cin >> n;

    int m = log2(n), mx = -1, dx = -1;

    for(int i = m; i >= 1; --i) {

        ll sum = 0;
        int idx = 1;
        bool flag = false;
        while(sum < n) {
            sum += pow(idx, i);

            if(sum == n) {
                flag = true;
                break;
            }
            idx++;
        }

        if(flag) {
            mx = i, dx = idx;
            break;
        }

    }

    if(mx == -1 && dx == -1) {
        cout << "Impossible for " << n << ".\n";
    } else {
        for(int i = 1; i <= dx; ++i) {
            cout << i << '^' << mx;
            if(i != dx) cout << '+';
        }
        cout << '\n';
    }

    return 0;
}
```

L1-8 现代战争（模拟）

```
#include <bits/stdc++.h>
using namespace std;

vector<bool> heng, lie;

int main() {

    int n, m, k;
    cin >> n >> m >> k;
    vector<vector<int>> v(n, vector<int>(m));
    vector<bool> derow(n, false);
    vector<bool> decol(m, false);
    for(int i = 0; i < n; ++i)
        for(int j = 0; j < m; ++j) {
            cin >> v[i][j];
        }

    while(k--) {

        int mx = INT_MIN;
        int im = -1, jm = -1;
        for(int i = 0; i < n; ++i) {
            if(derow[i]) continue;
            for(int j = 0; j < m; ++j) {
                if(decol[j]) continue;
                if(mx < v[i][j]) {
                    mx = v[i][j];
                    im = i, jm = j;
                }
            }
        }

        if(im != -1 && jm != -1) {
            derow[im] = true;
            decol[jm] = true;
        }

    }

    vector<vector<int>> res;
    for(int i = 0; i < n; ++i) {
        if(derow[i]) continue;
        vector<int> curow;
        for(int j = 0; j < m; ++j) {
            if(decol[j]) continue;
            curow.push_back(v[i][j]);
        }
        if(!curow.empty()) res.push_back(curow);
    }

    for(size_t i = 0; i < res.size(); ++i) {
        for(size_t j = 0; j < res[i].size(); ++j) {
            if (j > 0) cout << " ";
            cout << res[i][j];
        }
        cout << "\n";
    }

    return 0;
}
```

L2-1 算式拆解（字符串模拟栈操作）

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {

    stack<char> st;
    string s;
    cin >> s;

    for(size_t i = 0; i < s.size(); ++i) {
        if(s[i] == ')') {
            string ss = "";
            while(st.top() != '(') {
                ss += st.top();
                st.pop();
            }
            reverse(ss.begin(), ss.end());
            cout << ss;
            cout << '\n';
            st.pop();
        } else st.push(s[i]);
    }

    return 0;
}
```

L2-2 三点共线

1. 坐标处理（bitset无法处理负数）
2. 结构体数组去重（重构==，sort + unique + erase）

```cpp
bool operator==(const Node& other) const {
    return (x0 == other.x0 && x1 == other.x1 && x2 == other.x2);
}
sort(res.begin(), res.end());
auto last = unique(res.begin(), res.end());
res.erase(last, res.end());
```

```cpp
#include <bits/stdc++.h>
using namespace std;

const int MAXN = 1e6 + 5;

struct Node {
    int x0, x1, x2;

    bool operator==(const Node& other) const {
        return (x0 == other.x0 && x1 == other.x1 && x2 == other.x2);
    }
    bool operator<(const Node& other) const {
        if (x1 != other.x1) return x1 < other.x1;
        return x0 < other.x0;
    }
};

int main() {

    int n;
    cin >> n;

    bitset<6 * MAXN> y00, y11, y22;
    vector<int> y0, y1;

    for(int i = 0; i < n; ++i) {
        int x, y;
        cin >> x >> y;

        if(y == 0) {
            if(!y00.test(x + MAXN)) {
                y00.set(x + MAXN);
                y0.push_back(x);
            }
        } else if(y == 1) {
            if(!y11.test(x + MAXN)) {
                y11.set(x + MAXN);
                y1.push_back(x);
            }
        } else if(y == 2) {
            y22.set(x + 3 * MAXN);
        }
    }

    vector<Node> res;
    for(int x1 : y1) {
        for(int x0 : y0) {
            int x2 = 2 * x1 - x0;
            if(y22.test(x2 + 3 * MAXN)) {
                res.push_back({x0, x1, x2});
            }
        }
    }

    if(res.empty()) {
        cout << "-1\n";
    } else {
        // ！！！vector(结构体)去重！！！
        sort(res.begin(), res.end());
        auto last = unique(res.begin(), res.end());
        res.erase(last, res.end());

        for(auto [x0, x1, x2] : res) {
            printf("[%d, 0] [%d, 1] [%d, 2]\n", x0, x1, x2);
        }
    }

    return 0;
}
```

L2-3 胖达的山头（差分处理区间修改）

```
#include <bits/stdc++.h>
using namespace std;

int main() {

    int n;
    cin >> n;

    const int MAXN = 86400;
    vector<int> diff(MAXN + 2, 0);

    for(int i = 0; i < n; ++i) {
        int sh, sm, ss, eh, em, es;
        char c;
        cin >> sh >> c >> sm >> c >> ss;
        cin >> eh >> c >> em >> c >> es;

        int s = sh * 3600 + sm * 60 + ss;
        int e = eh * 3600 + em * 60 + es;

        diff[s] += 1;
        if(e + 1 <= MAXN) {
            diff[e + 1] -= 1;
        }
    }

    int mx = 0;
    int cur = 0;
    for(int i = 0; i < MAXN; ++i) {
        cur += diff[i];
        mx = max(mx, cur);
    }

    cout << mx;

    return 0;
}
```

L2-4 被n整除的n位数

先dfs进行构造，再判断是否处于区间中

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

int n;
ll a, b;
vector<ll> res;

// pre:前step位的数字, step:位数
void dfs(ll pre, int step) {

    if(step > n) {
        if(pre >= a && pre <= b) res.push_back(pre);
        return ;
    }

    // 最高位不能是0
    int start = (step == 1) ? 1 : 0;
    for(int d = start; d <= 9; ++d) {
        ll cur = pre * 10 + d;

        if(cur % step == 0) {
            dfs(cur, step + 1);
        }
    }
}

int main() {

    cin >> n >> a >> b;

    a = max(a, (ll)pow(10, n - 1));
    b = min(b, (ll)(pow(10, n) - 1));

    dfs(0, 1);

    if(res.empty()) {
        cout << "No Solution\n";
    } else {
        for(ll r : res) cout << r << '\n';
    }

    return 0;
}
```

L3-1 人生就像一场旅行（双指标FLoyd）

```cpp
#include <bits/stdc++.h>
using namespace std;

const int INF = 0x3f3f3f3f;
const int MAXN = 505;

int main() {

    int b, n, m, k;
    cin >> b >> n >> m >> k;

    // 最少旅费
    vector<vector<int>> dist(MAXN, vector<int>(MAXN, INF));
    // 最大心情指数
    vector<vector<int>> mood(MAXN, vector<int>(MAXN, 0));
    for(int i = 1; i <= n; ++i) {
        dist[i][i] = 0;
        mood[i][i] = 0;
    }

    for(int i = 0; i < m; ++i) {
        int u, v, cost, add;
        cin >> u >> v >> cost >> add;

        if(cost < dist[u][v]) {
            dist[u][v] = cost;
            dist[v][u] = cost;
            mood[u][v] = add;
            mood[v][u] = add;
        } else if(cost == dist[u][v]) {
            mood[u][v] = max(mood[u][v], add);
            mood[v][u] = max(mood[v][u], add);
        }
    }

    for(int k = 1; k <= n; ++k) {           // 中间点
        for(int i = 1; i <= n; ++i) {       // 起点
            for(int j = 1; j <= n; ++j) {   // 终点
                if(dist[i][k] == INF || dist[k][j] == INF) continue;
                if(dist[i][k] + dist[k][j] < dist[i][j]) {
                    dist[i][j] = dist[i][k] + dist[k][j];
                    mood[i][j] = mood[i][k] + mood[k][j];
                } else if(dist[i][k] + dist[k][j] == dist[i][j]) {
                    mood[i][j] = max(mood[i][k] + mood[k][j], mood[i][j]);
                }
            }
        }
    }

    while(k--) {
        int query;
        cin >> query;

        vector<int>reach;
        for(int j = 1; j <= n; ++j) {
            if(j == query) continue;
            if(dist[query][j] <= b) reach.push_back(j);
        }

        if(reach.empty()) {
            cout << "T_T\n";
        } else {
            sort(reach.begin(), reach.end());
            for(size_t i = 0; i < reach.size(); ++i) {
                if(i) cout << ' ';
                cout << reach[i];
            }
            cout << '\n';

            int mx = -1;
            vector<int> best;
            for(size_t i = 0; i < reach.size(); ++i) {
                if(mood[query][reach[i]] > mx) {
                    mx = mood[query][reach[i]];
                    best.clear();
                    best.push_back(reach[i]);
                } else if(mood[query][reach[i]] == mx) {
                    best.push_back(reach[i]);
                }
            }

            sort(best.begin(), best.end());
            for(size_t i = 0; i < best.size(); ++i) {
                if(i) cout << ' ';
                cout << best[i];
            }
            cout << '\n';
        }
    }

    return 0;
}
```

L3-2 影响力（暴力有部分分 20/30）

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

int main() {

    int n, m;
    cin >> n >> m;
    vector<vector<int>> v(n + 1, vector<int>(m + 1));
    vector<vector<ll>> res(n + 1, vector<ll>(m + 1, 0LL));

    for(int i = 1; i <= n; ++i) {
        for(int j = 1; j <= m; ++j) {
            cin >> v[i][j];
        }
    }

    for(int i = 1; i <= n; ++i) {
        for(int j = 1; j <= m; ++j) {

            for(int p = 1; p <= n; ++p) {
                for(int q = 1; q <= m; ++q) {
                    res[i][j] += (ll)v[i][j] * (ll)max(abs(i - p), abs(j - q));
                }
            }

        }
    }

    for(int i = 1; i <= n; ++i) {
        for(int j = 1; j <= m; ++j) {
            if(j != 1) cout << ' ';
            cout << res[i][j];
        }
        cout << '\n';
    }

    return 0;
}
```

L3-3 污染大亨

树形DP不会QAQ