

2025

L1-1 珍惜生命 ("Hello World!")

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

L2-2 偷感好重 (简单运算)

```
1 | #include <bits/stdc++.h>  
2 | using namespace std;  
3 |  
4 | int main() {  
5 |  
6 |     int a, b, c;  
7 |     cin >> a >> b >> c;  
8 |     cout << a + b + c;  
9 |  
10 |    return 0;  
11 | }
```

L1-3 高温补贴 (if - else)

```
1 | #include <bits/stdc++.h>  
2 | using namespace std;  
3 |  
4 | int main() {  
5 |  
6 |     int T, s, t;  
7 |     cin >> T >> s >> t;  
8 |  
9 |     if(T >= 35 && s == 1 && t >= 33) {  
10 |         cout << "Bu Tie\n";  
11 |         cout << T << '\n';  
12 |     } else if(s == 0 && T >= 35 && t >= 33) {  
13 |         cout << "Shi Nei\n";  
14 |         cout << T << '\n';  
15 |     } else if(s == 1) {  
16 |         cout << "Bu Re\n";  
17 |         cout << t << '\n';  
18 |     } else {  
19 |         cout << "Shu Shi\n";  
20 |         cout << t << '\n';  
21 |     }  
22 |  
23 |     return 0;  
24 | }
```

L1-4 零头就抹了吧 (简单while)

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

```

3
4 int main() {
5
6     int n;
7     cin >> n;
8
9     int i = 1;
10    while(i <= n) {
11        i *= 2;
12    }
13    cout << i / 2;
14
15    return 0;
16 }
```

L1-5 这是字符串题 (简单数组计数)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 string s;
5 int f[26], cnt[26];
6
7 int main() {
8
9
10    cin >> s;
11    for(int i = 0; i < 26; ++i) cin >> f[i];
12
13    int sum = 0;
14    for(size_t i = 0; i < s.size(); ++i) {
15        sum += f[s[i] - 'a'];
16        cnt[s[i] - 'a']++;
17    }
18
19    for(int i = 0; i < 26; ++i) {
20        if(i) cout << ' ';
21        cout << cnt[i];
22    }
23    cout << '\n';
24    cout << sum;
25
26    return 0;
27 }
```

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);
```

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5
6     int n, m;
7     cin >> n >> m;
8     string s1 = "";
9     for(int i = 0; i < n; ++i) {
10         int x;
11         cin >> x;
12         s1 += (char)(x - 1 + 'a');
13     }
14
15     while(m--) {
16
17         int op;
18         cin >> op;
19
20         if(op == 1) {
21
22             int a1, a2;
23             string b1 = "", b2 = "";
24
25             cin >> a1;
26             for(int i = 0; i < a1; ++i) {
27                 int x;
28                 cin >> x;
29                 b1 += (char)(x - 1 + 'a');
30             }
31
32             cin >> a2;
33             for(int i = 0; i < a2; ++i) {
34                 int x;
35                 cin >> x;
36                 b2 += (char)(x - 1 + 'a');
37             }
38
39             if(s1.find(b1) != string::npos) {
40                 s1.replace(s1.find(b1), a1, b2);
41             }
42
43         } else if(op == 2) {
44
45             for(size_t i = 1; i < s1.size(); ++i) {
46                 int val = (s1[i - 1] - 'a') + (s1[i] - 'a');
47                 if(val % 2 == 0) {
```

```

48             char ch = (char)(val / 2 + 'a');
49             s1.insert(i, 1, ch);
50             i++;
51         }
52     }
53
54 } else if(op == 3) {
55     int l, r;
56     cin >> l >> r;
57     reverse(s1.begin() + (l - 1), s1.begin() + r);
58 }
59
60 }
61
62 for(size_t i = 0; i < s1.size(); ++i) {
63     if(i != 0) cout << ' ';
64     cout << (int)(s1[i] - 'a' + 1);
65 }
66
67 return 0;
68 }
```

L1-7 大幂数 (暴力)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 #define ll long long
4
5 int main() {
6
7     ll n;
8     cin >> n;
9
10    int m = log2(n), mx = -1, dx = -1;
11
12    for(int i = m; i >= 1; --i) {
13
14        ll sum = 0;
15        int idx = 1;
16        bool flag = false;
17        while(sum < n) {
18            sum += pow(idx, i);
19
20            if(sum == n) {
21                flag = true;
22                break;
23            }
24            idx++;
25        }
26
27        if(flag) {
28            mx = i, dx = idx;
29            break;
30        }
31    }
32 }
```

```

32     }
33
34     if(mx == -1 && dx == -1) {
35         cout << "Impossible for " << n << ".\n";
36     } else {
37         for(int i = 1; i <= dx; ++i) {
38             cout << i << '^' << mx;
39             if(i != dx) cout << '+';
40         }
41         cout << '\n';
42     }
43
44     return 0;
45 }
```

L1-8 现代战争 (模拟)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 vector<bool> heng, lie;
5
6 int main() {
7
8     int n, m, k;
9     cin >> n >> m >> k;
10    vector<vector<int>> v(n, vector<int>(m));
11    vector<bool> derow(n, false);
12    vector<bool> decol(m, false);
13    for(int i = 0; i < n; ++i)
14        for(int j = 0; j < m; ++j) {
15            cin >> v[i][j];
16        }
17
18    while(k--) {
19
20        int mx = INT_MIN;
21        int im = -1, jm = -1;
22        for(int i = 0; i < n; ++i) {
23            if(derow[i]) continue;
24            for(int j = 0; j < m; ++j) {
25                if(decol[j]) continue;
26                if(mx < v[i][j]) {
27                    mx = v[i][j];
28                    im = i, jm = j;
29                }
30            }
31        }
32
33        if(im != -1 && jm != -1) {
34            derow[im] = true;
35            decol[jm] = true;
36        }
37
38    }
}
```

```

39     vector<vector<int>> res;
40     for(int i = 0; i < n; ++i) {
41         if(derow[i]) continue;
42         vector<int> curow;
43         for(int j = 0; j < m; ++j) {
44             if(decol[j]) continue;
45             curow.push_back(v[i][j]);
46         }
47     }
48     if(!curow.empty()) res.push_back(curow);
49 }
50
51 for(size_t i = 0; i < res.size(); ++i) {
52     for(size_t j = 0; j < res[i].size(); ++j) {
53         if (j > 0) cout << " ";
54         cout << res[i][j];
55     }
56     cout << "\n";
57 }
58
59 return 0;
60 }
```

L2-1 算式拆解 (字符串模拟栈操作)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5
6     stack<char> st;
7     string s;
8     cin >> s;
9
10    for(size_t i = 0; i < s.size(); ++i) {
11        if(s[i] == ')') {
12            string ss = "";
13            while(st.top() != '(') {
14                ss += st.top();
15                st.pop();
16            }
17            reverse(ss.begin(), ss.end());
18            cout << ss;
19            cout << '\n';
20            st.pop();
21        } else st.push(s[i]);
22    }
23
24    return 0;
25 }
```

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)

```

1  bool operator==(const Node& other) const {
2      return (x0 == other.x0 && x1 == other.x1 && x2 == other.x2);
3  }
4  sort(res.begin(), res.end());
5  auto last = unique(res.begin(), res.end());
6  res.erase(last, res.end());

```

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int MAXN = 1e6 + 5;
5
6 struct Node {
7     int x0, x1, x2;
8
9     bool operator==(const Node& other) const {
10         return (x0 == other.x0 && x1 == other.x1 && x2 == other.x2);
11     }
12     bool operator<(const Node& other) const {
13         if (x1 != other.x1) return x1 < other.x1;
14         return x0 < other.x0;
15     }
16 };
17
18 int main() {
19
20     int n;
21     cin >> n;
22
23     bitset<6 * MAXN> y00, y11, y22;
24     vector<int> y0, y1;
25
26     for(int i = 0; i < n; ++i) {
27         int x, y;
28         cin >> x >> y;
29
30         if(y == 0) {
31             if(!y00.test(x + MAXN)) {
32                 y00.set(x + MAXN);
33                 y0.push_back(x);
34             }
35         } else if(y == 1) {
36             if(!y11.test(x + MAXN)) {
37                 y11.set(x + MAXN);
38                 y1.push_back(x);
39             }
40         } else if(y == 2) {
41             y22.set(x + 3 * MAXN);
42         }
43     }
44
45     vector<Node> res;
46     for(int x1 : y1) {
47         for(int x0 : y0) {

```

```

48         int x2 = 2 * x1 - x0;
49         if(y22.test(x2 + 3 * MAXN)) {
50             res.push_back({x0, x1, x2});
51         }
52     }
53 }
54
55 if(res.empty()) {
56     cout << "-1\n";
57 } else {
58     // !!! vector(结构体)去重!!!
59     sort(res.begin(), res.end());
60     auto last = unique(res.begin(), res.end());
61     res.erase(last, res.end());
62
63     for(auto [x0, x1, x2] : res) {
64         printf("[%d, 0] [%d, 1] [%d, 2]\n", x0, x1, x2);
65     }
66 }
67
68 return 0;
69 }
```

L2-3 胖达的山头 (差分处理区间修改)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5
6     int n;
7     cin >> n;
8
9     const int MAXN = 86400;
10    vector<int> diff(MAXN + 2, 0);
11
12    for(int i = 0; i < n; ++i) {
13        int sh, sm, ss, eh, em, es;
14        char c;
15        cin >> sh >> c >> sm >> c >> ss;
16        cin >> eh >> c >> em >> c >> es;
17
18        int s = sh * 3600 + sm * 60 + ss;
19        int e = eh * 3600 + em * 60 + es;
20
21        diff[s] += 1;
22        if(e + 1 <= MAXN) {
23            diff[e + 1] -= 1;
24        }
25    }
26
27    int mx = 0;
28    int cur = 0;
29    for(int i = 0; i < MAXN; ++i) {
30        cur += diff[i];
31
32        if(cur > mx) {
33            mx = cur;
34        }
35    }
36
37    cout << mx << endl;
38 }
```

```

31         mx = max(mx, cur);
32     }
33
34     cout << mx;
35
36     return 0;
37 }
```

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

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 #define ll long long
4
5 int n;
6 ll a, b;
7 vector<ll> res;
8
9 // pre:前step位的数字, step:位数
10 void dfs(ll pre, int step) {
11
12     if(step > n) {
13         if(pre >= a && pre <= b) res.push_back(pre);
14         return ;
15     }
16
17     // 最高位不能是0
18     int start = (step == 1) ? 1 : 0;
19     for(int d = start; d <= 9; ++d) {
20         ll cur = pre * 10 + d;
21
22         if(cur % step == 0) {
23             dfs(cur, step + 1);
24         }
25     }
26 }
27
28 int main() {
29
30     cin >> n >> a >> b;
31
32     a = max(a, (ll)pow(10, n - 1));
33     b = min(b, (ll)(pow(10, n) - 1));
34
35     dfs(0, 1);
36
37     if(res.empty()) {
38         cout << "No Solution\n";
39     } else {
40         for(ll r : res) cout << r << '\n';
41     }
42
43     return 0;
44 }
```

L3-1 人生就像一场旅行 (双指标FLoyd)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 const int INF = 0x3f3f3f3f;
5 const int MAXN = 505;
6
7 int main() {
8
9     int b, n, m, k;
10    cin >> b >> n >> m >> k;
11
12    // 最少旅费
13    vector<vector<int>> dist(MAXN, vector<int>(MAXN, INF));
14    // 最大心情指数
15    vector<vector<int>> mood(MAXN, vector<int>(MAXN, 0));
16    for(int i = 1; i <= n; ++i) {
17        dist[i][i] = 0;
18        mood[i][i] = 0;
19    }
20
21    for(int i = 0; i < m; ++i) {
22        int u, v, cost, add;
23        cin >> u >> v >> cost >> add;
24
25        if(cost < dist[u][v]) {
26            dist[u][v] = cost;
27            dist[v][u] = cost;
28            mood[u][v] = add;
29            mood[v][u] = add;
30        } else if(cost == dist[u][v]) {
31            mood[u][v] = max(mood[u][v], add);
32            mood[v][u] = max(mood[v][u], add);
33        }
34    }
35
36    for(int k = 1; k <= n; ++k) {           // 中间点
37        for(int i = 1; i <= n; ++i) {       // 起点
38            for(int j = 1; j <= n; ++j) {   // 终点
39                if(dist[i][k] == INF || dist[k][j] == INF) continue;
40                if(dist[i][k] + dist[k][j] < dist[i][j]) {
41                    dist[i][j] = dist[i][k] + dist[k][j];
42                    mood[i][j] = mood[i][k] + mood[k][j];
43                } else if(dist[i][k] + dist[k][j] == dist[i][j]) {
44                    mood[i][j] = max(mood[i][k] + mood[k][j], mood[i][j]);
45                }
46            }
47        }
48    }
49
50    while(k--) {
51        int query;

```

```

52     cin >> query;
53
54     vector<int>reach;
55     for(int j = 1; j <= n; ++j) {
56         if(j == query) continue;
57         if(dist[query][j] <= b) reach.push_back(j);
58     }
59
60     if(reach.empty()) {
61         cout << "T_T\n";
62     } else {
63         sort(reach.begin(), reach.end());
64         for(size_t i = 0; i < reach.size(); ++i) {
65             if(i) cout << ' ';
66             cout << reach[i];
67         }
68         cout << '\n';
69
70         int mx = -1;
71         vector<int> best;
72         for(size_t i = 0; i < reach.size(); ++i) {
73             if(mood[query][reach[i]] > mx) {
74                 mx = mood[query][reach[i]];
75                 best.clear();
76                 best.push_back(reach[i]);
77             } else if(mood[query][reach[i]] == mx) {
78                 best.push_back(reach[i]);
79             }
80         }
81
82         sort(best.begin(), best.end());
83         for(size_t i = 0; i < best.size(); ++i) {
84             if(i) cout << ' ';
85             cout << best[i];
86         }
87         cout << '\n';
88     }
89 }
90
91 return 0;
92 }
```

L3-2 影响力 (暴力有部分分 20/30)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 #define ll long long
4
5 int main() {
6
7     int n, m;
8     cin >> n >> m;
9     vector<vector<int>> v(n + 1, vector<int>(m + 1));
10    vector<vector<ll>> res(n + 1, vector<ll>(m + 1, 0LL));
11
```

```
12     for(int i = 1; i <= n; ++i) {
13         for(int j = 1; j <= m; ++j) {
14             cin >> v[i][j];
15         }
16     }
17
18     for(int i = 1; i <= n; ++i) {
19         for(int j = 1; j <= m; ++j) {
20
21             for(int p = 1; p <= n; ++p) {
22                 for(int q = 1; q <= m; ++q) {
23                     res[i][j] += (ll)v[i][j] * (ll)max(abs(i - p), abs(j -
q));
24                 }
25             }
26         }
27     }
28 }
29
30     for(int i = 1; i <= n; ++i) {
31         for(int j = 1; j <= m; ++j) {
32             if(j != 1) cout << ' ';
33             cout << res[i][j];
34         }
35         cout << '\n';
36     }
37
38     return 0;
39 }
```

L3-3 污染大亨

树形DP不会QAQ