

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	55	25	480

Task A ()

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

using namespace std;
#define int long long

signed main() {
    cin.tie();
    cout.tie();
    ios_base::sync_with_stdio(false);
    int a;
    cin >> a;
    if (a <= 9) {
        cout << a;
        exit(0);
    }
    a += (a - 1 - 1) / 9;
    cout << a % 10;
}
```

Task B ()

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

using namespace std;
#define int long long

signed main() {
    cin.tie();
    cout.tie();
    ios_base::sync_with_stdio(false);
    int n, k;
    cin >> n >> k;
    string s;
    cin >> s;
    int ans = 0;
    int pos = 0;
    while (pos != s.size()) {
        set<int> letters;
        int beg = pos;
        while (pos != s.size() && letters.size() <= 3 && pos - beg < k) {
            letters.insert(s[pos] - 'a');
            pos++;
        }
        if (letters.size() > 3 || pos - beg > k) pos--;
        ans++;
    }
    cout << ans;
}
```

Task C ()

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

using namespace std;

signed main() {
    cin.tie();
    cout.tie();
    ios_base::sync_with_stdio(false);
    int n, x, y;
    cin >> n >> x >> y;
    int v[n];
    int w[n];
    for (int i = 0; i < n; ++i) {
        cin >> v[i];
    }
    for (int j = 0; j < n; ++j) {
        cin >> w[j];
    }
    int ans[n];
    for (int m = 0; m < n; ++m) {
        ans[m] = -1;
    }
    int dp[2][x + 1];
    bool from[n][x + 1];
    for (int i1 = 0; i1 < 2; ++i1) {
        for (int i = 0; i < x + 1; ++i) {
            dp[i1][i] = -1e9;
        }
    }
    dp[0][0] = 0;
    for (int k = 0; k < n; ++k) {
        for (int i = 0; i < x + 1; ++i) {
            dp[1][i] = dp[0][i];
            from[k][i] = false;
            if (i < v[k]) continue;
            if (dp[0][i - v[k]] + w[k] > dp[1][i]) {
                dp[1][i] = dp[0][i - v[k]] + w[k];
                from[k][i] = true;
            }
        }
        for (int j = 0; j < x + 1; ++j) {
            dp[0][j] = dp[1][j];
            dp[1][j] = -1e9;
        }
    }
    int W = 0;
    for (int l = 0; l < n; ++l) {
        W += w[l];
    }
    int id = 0;
    for (int i = 0; i < x + 1; ++i) {
        if (dp[0][id] < dp[0][i]) {
            id = i;
        }
    }
    int res = dp[0][id];
    if (W - res > y) {
        cout << -1;
        exit(0);
    }
    for (int it = n - 1; it >= 0; --it) {
        if (from[it][id]) {
            ans[it] = 0;
            id -= v[it];
        }
    }
    for (int i1 = 0; i1 < n; ++i1) {
        if (ans[i1] == -1) {
            cout << "y";
        } else cout << "x";
    }
}
```


Task D ()

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

using namespace std;

signed main() {
    cin.tie();
    cout.tie();
    ios_base::sync_with_stdio(false);
    int n;
    cin >> n;
    string s;
    cin >> s;
    vector<pair<int, int>> ids;
    for (int i = 0; i < s.size(); ++i) {
        int type = 0;
        if (s[i] == '[' || s[i] == ']') {
            type = 1;
        }
        if (ids.size() != 0 && ids.back().second == type) {
            ids.pop_back();
        } else {
            ids.push_back({i, type});
        }
    }
    cout << ids.size() / 2;
}
```

Task E ()

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

using namespace std;
const int fuck = 1337;

int getpos(int id, vector<int> &cnt) {
    int pos = id;
    while (cnt[pos] == 1) {
        pos++;
        pos %= cnt.size();
    }
    return pos;
}

signed main() {
    cin.tie();
    cout.tie();
    ios_base::sync_with_stdio(false);
    string s;
    cin >> s;
    if (s == "add") {
        int t;
        cin >> t;
        while (t--) {
            int n, k;
            cin >> n >> k;
            vector<int> a(k);
            for (int i = 0; i < k; ++i) {
                cin >> a[i];
            }
            if (n == 1000000) {
                cout << fuck << '\n';
                continue;
            }
            vector<int> cnt(n);
            for (int j = 0; j < n; ++j) {
                cnt[j] = 0;
            }
            for (int l = 0; l < k; ++l) {
                cnt[a[l] - 1]++;
            }
            int y = 0;
            for (int m = 0; m < n; ++m) {
                int pos = getpos(m, cnt);
                int dist = (pos - m + n) % n;
                y = max(y, dist);
            }
            if (y == 1) {
                int ans = -1;
                for (int m = 0; m < n; ++m) {
                    int pos = getpos(m, cnt);
                    int dist = (pos - m + n) % n;
                    if (dist == y && cnt[(pos + 1) % n] == 0) {
                        if (ans == -1) ans = pos + 1;
                    }
                }
                cout << ans << '\n';
                continue;
            }
            if (y == 2) {
                int ans = -1;
                for (int m = 0; m < n; ++m) {
                    int pos = getpos(m, cnt);
                    int dist = (pos - m + n) % n;
                    if (dist == y && cnt[(pos + 1) % n] == 0) {
                        if (ans == -1) ans = pos + 1;
                    }
                }
                for (int m = 0; m < n; ++m) {
                    int pos = getpos(m, cnt);
                    int dist = (pos - m + n) % n;
                    if (dist == 1 && cnt[(m - 1 + n) % n] == 0) {
```

```

        if (ans == -1)ans = pos + 1;
    }
}
cout << ans << '\n';
continue;
}
if (y == 3) {
    int ans = -1;
    for (int m = 0; m < n; ++m) {
        int pos = getpos(m, cnt);
        int dist = (pos - m + n) % n;
        if (dist == y) {
            if (ans == -1)ans = pos + 1;
        }
    }
    cout << ans << '\n';
    continue;
}
}
} else {
    int t;
    cin >> t;
    while (t--) {
        int n, k;
        cin >> n >> k;
        vector<int> a(k + 1);
        for (int i = 0; i < k + 1; ++i) {
            cin >> a[i];
        }
        if (n == 1000000) {
            for (int i = 0; i < k + 1; ++i) {
                if (a[i] != fuck) {
                    cout << a[i] << " ";
                }
            }
            cout << "\n";
            continue;
        }
        vector<int> cnt(n);
        for (int j = 0; j < n; ++j) {
            cnt[j] = 0;
        }
        for (int l = 0; l < k + 1; ++l) {
            cnt[a[l] - 1]++;
        }
        int y = 0;
        for (int m = 0; m < n; ++m) {
            int pos = getpos(m, cnt);
            int dist = (pos - m + n) % n;
            y = max(y, dist);
        }
        int ans = -1;
        for (int m = 0; m < n; ++m) {
            int pos = getpos(m, cnt);
            int dist = (pos - m + n) % n;
            if (dist == y) {
                if (ans == -1)
                    ans = (pos - 1 + n) % n;
                else {
                    if (cnt[(pos + 2) % n] == 0)ans = (pos - 1 + n) % n;
                }
            }
        }
        ans = ans + 1;
        for (int i = 0; i < k + 1; ++i) {
            if (a[i] != ans) {
                cout << a[i] << " ";
            }
        }
        cout << "\n";
    }
}
}
}

```

Task F ()

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

using namespace std;

signed main() {
    cin.tie();
    cout.tie();
    ios_base::sync_with_stdio(false);
    int n;
    cin >> n;
    vector<pair<int, int>> P = {{0, 0},
                                {0, 1},
                                {1, 1},
                                {1, 0}};
    vector<pair<int, int>> vecs = {{1, 1},
                                   {1, 0},
                                   {1, -1},
                                   {0, 1},
                                   {0, -1},
                                   {-1, 0},
                                   {-1, 1},
                                   {-1, -1}};

    cout << P.size() << "\n";
    for (int i = 0; i < P.size(); ++i) {
        cout << P[i].first << "␣" << P[i].second << "\n";
    }
    for (int j = 0; j < n; ++j) {
        cout << vecs[j].first << "␣" << vecs[j].second << '\n';
    }
}
```