Олимпиада СПбГУ по информатике 2020/21 учебного года

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

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <assert.h>
#include <cassert>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <math.h>
#include <string>

using namespace std;

#define int long long

#define ll long long
#define ld long double

#define pii pair<int, int>
#define mp make_pair
#define ft first
#define sd second

#define all(a) a.begin(), a.end()
#define mysort(a) sort(all(a))
#define myreverse(a) reverse(all(a))
#define myunique(a) unique(all(a))
#define superunique(a) a.resize(myunique(a) - a.begin())

#define FOR(i, l, r) for (int i = l; i < r; i++)
#define FORD(i, r, l) for (int i = r; i >=l; i--)



template <class T>
istream& operator>>(istream & in, vector<T>&vec) {
        for (int i = 0; i < vec.size(); i++)
                in >> vec[i];
        return in;
}

template <class T1, class T2>
istream& operator>>(istream& in, pair<T1, T2>& pr) {
        return in >> pr.ft >> pr.sd;
}

template <class T>
ostream& operator<<(ostream& out, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                out << vec[i] << '␣';
        return out << '\n';
}
```

```
template <class T1, class T2>
ostream& operator<<(ostream& out, pair<T1, T2>& pr) {
        return out << pr.ft << '␣' << pr.sd;
}

void solve() {
        int k;
        cin >> k;
        if (k <= 9)
                cout << k;
        else {
                vector<int> ans = {0, 2, 3, 4 , 5, 6, 7, 8, 9};
                cout << ans[(k - 10) % 9];
        }
}

int32_t main() {
        cin.tie(0);
        ios_base::sync_with_stdio(0);
        int t = 1;
        //freopen("in.txt", "r", stdin);
        //freopen("out.txt", "w", stdout);
        //cin >> t;
        while (t--)
                solve();
}
```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <assert.h>
#include <cassert>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <math.h>
#include <string>

using namespace std;

#define int long long

#define ll long long
#define ld long double

#define pii pair<int, int>
#define mp make_pair
#define ft first
#define sd second

#define all(a) a.begin(), a.end()
#define mysort(a) sort(all(a))
#define myreverse(a) reverse(all(a))
#define myunique(a) unique(all(a))
#define superunique(a) a.resize(myunique(a) - a.begin())

#define FOR(i, l, r) for (int i = l; i < r; i++)
#define FORD(i, r, l) for (int i = r; i >=l; i--)



template <class T>
istream& operator>>(istream & in, vector<T>&vec) {
        for (int i = 0; i < vec.size(); i++)
                in >> vec[i];
        return in;
}

template <class T1, class T2>
istream& operator>>(istream& in, pair<T1, T2>& pr) {
        return in >> pr.ft >> pr.sd;
}

template <class T>
ostream& operator<<(ostream& out, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                out << vec[i] << '␣';
        return out << '\n';
}

template <class T1, class T2>
ostream& operator<<(ostream& out, pair<T1, T2>& pr) {
        return out << pr.ft << '␣' << pr.sd;
}

void solve() {
        int n, k;
        cin >> n >> k;
        vector<vector<int>> prefix('z' - 'a' + 1, vector<int>(n + 1));
        string s;
        cin >> s;
        FOR(i, 0, n) {
                for (int c = 'a'; c <= 'z'; c++)
                        prefix[c - 'a'][i + 1] = prefix[c - 'a'][i] + (s[i] == c);
        }
        int i = 0;
        int ans = 0;
        while (i < n) {
                int l = 0;
                int r = min(k, n - i) + 1;
                while (r - l > 1) {
                        int m = (l + r) / 2;
                        int cnt = 0;
                        for (int c = 0; c <= 'z' - 'a'; c++)
                                if (prefix[c][i + m] - prefix[c][i] > 0)
                                        cnt++;
                        if (cnt > 3)
                                r = m;
                        else
                                l = m;
                }
                i = i + l;
                ans++;
        }
        cout << ans;
}

int32_t main() {
        cin.tie(0);
        ios_base::sync_with_stdio(0);
        int t = 1;
        //freopen("in.txt", "r", stdin);
        //freopen("out.txt", "w", stdout);
        //cin >> t;
        while (t--)
                solve();
}
```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <assert.h>
#include <cassert>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <math.h>
#include <string>

using namespace std;

#define ll long long
#define ld long double

#define pii pair<int, int>
#define mp make_pair
#define ft first
#define sd second

#define all(a) a.begin(), a.end()
#define mysort(a) sort(all(a))
#define myreverse(a) reverse(all(a))
#define myunique(a) unique(all(a))
#define superunique(a) a.resize(myunique(a) - a.begin())

#define FOR(i, l, r) for (int i = l; i < r; i++)
#define FORD(i, r, l) for (int i = r; i >=l; i--)



template <class T>
istream& operator>>(istream & in, vector<T>&vec) {
        for (int i = 0; i < vec.size(); i++)
                in >> vec[i];
        return in;
}

template <class T1, class T2>
istream& operator>>(istream& in, pair<T1, T2>& pr) {
        return in >> pr.ft >> pr.sd;
}

template <class T>
ostream& operator<<(ostream& out, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                out << vec[i] << '␣';
        return out << '\n';
}

template <class T1, class T2>
ostream& operator<<(ostream& out, pair<T1, T2>& pr) {
        return out << pr.ft << '␣' << pr.sd;
}

void solve() {
        int n, maxv, maxw;
        cin >> n >> maxv >> maxw;
        vector<pair<pii, int>> items(n);
        vector<int> weights(n);
        for (int i = 0; i < n; i++)
                cin >> items[i].ft.ft;
        for (int i = 0; i < n; i++) {
                cin >> items[i].ft.sd;
                weights[i] = items[i].ft.sd;
                items[i].sd = i;
        }
        mysort(items);
```

```
    mysort(weights);
    int x = 0;
    int summ = 0;
    while (x < n && summ + weights[x] <= maxw) {
        summ += weights[x];
        x++;
    }
    int y = 0;
    summ = 0;
    while (y < n && summ + items[y].ft.ft <= maxv) {
        summ += items[y].ft.ft;
        y++;
    }
    if (x + y < n) {
        cout << -1;
        return;
    }
    vector<vector<bool>> p(n + 1);
    vector<vector<int>> dp(n + 1);
    dp[0].resize(1);
    p[0].resize(1);
    dp[0][0] = 0;
    int sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += items[i - 1].ft.ft;
        dp[i].resize(min(sum, maxv) + 1, maxw + 1);
        p[i].resize(min(sum, maxv) + 1, false);
    }
    for (int i = 0; i < n; i++) {
        for (int v = 0; v < dp[i].size() && v + items[i].ft.ft < dp[i + 1].size(); v++)
            if (dp[i + 1][v + items[i].ft.ft] > dp[i][v]) {
                dp[i + 1][v + items[i].ft.ft] = dp[i][v];
                p[i + 1][v + items[i].ft.ft] = false;
            }
        for (int v = 0; v < dp[i].size(); v++)
            if (dp[i + 1][v] > dp[i][v] + items[i].ft.sd) {
                dp[i + 1][v] = dp[i][v] + items[i].ft.sd;
                p[i + 1][v] = true;
            }
    }
    for (int v = 0; v < dp[n].size(); v++)
        if (dp[n][v] <= maxw) {
            vector<char> ans(n);
            int curv = v;
            for (int i = n; i > 0; i--)
                if (p[i][v])
                    ans[items[i - 1].sd] = 'y';
                else {
                    ans[items[i - 1].sd] = 'x';
                    v -= items[i - 1].ft.ft;
                }
            for (char c : ans)
                cout << c;
            return;
        }
    cout << -1;
    return;
}

int32_t main() {
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    int t = 1;
    //freopen("in.txt", "r", stdin);
    //freopen("out.txt", "w", stdout);
    //cin >> t;
    while (t--)
        solve();
}
```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <assert.h>
#include <cassert>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <math.h>
#include <string>

using namespace std;

#define ll long long
#define ld long double

#define pii pair<int, int>
#define mp make_pair
#define ft first
#define sd second

#define all(a) a.begin(), a.end()
#define mysort(a) sort(all(a))
#define myreverse(a) reverse(all(a))
#define myunique(a) unique(all(a))
#define superunique(a) a.resize(myunique(a) - a.begin())

#define FOR(i, l, r) for (int i = l; i < r; i++)
#define FORD(i, r, l) for (int i = r; i >=l; i--)



template <class T>
istream& operator>>(istream & in, vector<T>&vec) {
        for (int i = 0; i < vec.size(); i++)
                in >> vec[i];
        return in;
}

template <class T1, class T2>
istream& operator>>(istream& in, pair<T1, T2>& pr) {
        return in >> pr.ft >> pr.sd;
}

template <class T>
ostream& operator<<(ostream& out, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                out << vec[i] << '␣';
        return out << '\n';
}

template <class T1, class T2>
ostream& operator<<(ostream& out, pair<T1, T2>& pr) {
        return out << pr.ft << '␣' << pr.sd;
}

void solve() {
        int n;
        cin >> n;
        vector<bool> t;
        FOR(i, 0, 2 * n) {
                char c;
                cin >> c;
                if (c == '[' || c == ']')
                        t.push_back(1);
                else
                        t.push_back(0);
                while (t.size() > 1 && *--t.end() == *----t.end()) {
                        t.pop_back();
                        t.pop_back();
                }
        }
        cout << t.size() / 2;
}

int32_t main() {
        cin.tie(0);
        ios_base::sync_with_stdio(0);
        int t = 1;
        //freopen("in.txt", "r", stdin);
        //freopen("out.txt", "w", stdout);
        //cin >> t;
        while (t--)
                solve();
}
```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <assert.h>
#include <cassert>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <math.h>
#include <string>

using namespace std;

#define ll long long
#define ld long double

#define pii pair<int, int>
#define mp make_pair
#define ft first
#define sd second

#define all(a) a.begin(), a.end()
#define mysort(a) sort(all(a))
#define myreverse(a) reverse(all(a))
#define myunique(a) unique(all(a))
#define superunique(a) a.resize(myunique(a) - a.begin())

#define FOR(i, l, r) for (int i = l; i < r; i++)
#define FORD(i, r, l) for (int i = r; i >=l; i--)



template <class T>
istream& operator>>(istream& in, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                in >> vec[i];
        return in;
}

template <class T1, class T2>
istream& operator>>(istream& in, pair<T1, T2>& pr) {
        return in >> pr.ft >> pr.sd;
}

template <class T>
ostream& operator<<(ostream& out, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                out << vec[i] << '␣';
        return out << '\n';
}

template <class T1, class T2>
ostream& operator<<(ostream& out, pair<T1, T2>& pr) {
        return out << pr.ft << '␣' << pr.sd;
}

void solve() {
        vector<bool> used(1024, false);
        vector<int> p(1024), child(1024);
        FOR(a, 0, 10) {
                FOR(b, a + 1, 10) {
                        FOR(c, b + 1, 10) {
                                int mask1 = (1 << a) ^ (1 << b) ^ (1 << c);
                                bool found = false;
                                for (int d = 0; d < 10 && !found; d++)
                                        if (d != a && d != b && d != c) {
                                                if (!used[mask1 ^ (1 << d)]) {
                                                        p[mask1] = d;
                                                        child[mask1 ^ (1 << d)] = d;
                        found = true;
                        used[mask1 ^ (1 << d)] = true;
                    }
                }
            if (!found)
                int zero = 0;
        }
    }
}
string s;
cin >> s;
if (s[0] == 'a') {
    int t;
    cin >> t;
    while (t--) {
        int n, k;
        cin >> n >> k;
        if (n == 10) {
            int mask = 0;
            for (int i = 0; i < k; i++) {
                int x;
                cin >> x;
                mask ^= 1 << x - 1;
            }
            cout << p[mask] + 1 << '\n';
        }
        else if (n == 1000000) {
            vector<int> nums(k);
            cin >> nums;
            mysort(nums);
            int sum = 0;
            for (int num : nums)
                sum = (sum + num - 1) % (k + 1);
            while (true) {
                int x = rand() % n;
                bool good = true;
                auto it = lower_bound(all(nums), x);
                if (it != nums.end() && *it == x)
                    continue;
                else
                    if ((sum + x) % (k + 1) == it - nums.begin()) {
                        cout << x + 1;
                        break;
                    }
            }
        }
    }
}
else {
    int t;
    cin >> t;
    while (t--) {
        int n, k;
        cin >> n >> k;
        if (n == 10) {
            int mask = 0;
            for (int i = 0; i < k + 1; i++) {
                int x;
                cin >> x;
                mask ^= 1 << x - 1;
            }
            mask ^= 1 << child[mask];
            FOR(i, 0, 10)
                if ((mask >> i) % 2)
                    cout << i + 1 << '␣';
            cout << '\n';
        }
        else if (n == 1000000) {
            vector<int> nums(k + 1);
            cin >> nums;
            mysort(nums);
            int summ = 0;
            for (int i = 0; i < k + 1; i++)
                summ = (summ + nums[i] - 1) % (k + 1);
```

```
                                for (int num : nums)
                                        if (num != nums[summ])
                                                cout << num << '␣';
                                cout << '\n';
                        }
                }
        }
        int zero = 0;
}

int32_t main() {
        cin.tie(0);
        ios_base::sync_with_stdio(0);
        int t = 1;
        //freopen("in.txt", "r", stdin);
        //freopen("out.txt", "w", stdout);
        //cin >> t;
        while (t--)
                solve();
}
```

Task F ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <assert.h>
#include <cassert>
#include <algorithm>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <math.h>
#include <string>

using namespace std;

#define ll long long
#define ld long double

#define pii pair<int, int>
#define mp make_pair
#define ft first
#define sd second

#define all(a) a.begin(), a.end()
#define mysort(a) sort(all(a))
#define myreverse(a) reverse(all(a))
#define myunique(a) unique(all(a))
#define superunique(a) a.resize(myunique(a) - a.begin())

#define FOR(i, l, r) for (int i = l; i < r; i++)
#define FORD(i, r, l) for (int i = r; i >=l; i--)



template <class T>
istream& operator>>(istream & in, vector<T>&vec) {
        for (int i = 0; i < vec.size(); i++)
                in >> vec[i];
        return in;
}

template <class T1, class T2>
istream& operator>>(istream& in, pair<T1, T2>& pr) {
        return in >> pr.ft >> pr.sd;
}

template <class T>
ostream& operator<<(ostream& out, vector<T>& vec) {
        for (int i = 0; i < vec.size(); i++)
                out << vec[i] << ' ';
        return out << '\n';
}

template <class T1, class T2>
ostream& operator<<(ostream& out, pair<T1, T2>& pr) {
        return out << pr.ft << ' ' << pr.sd;
}

void solve() {
        int n;
        cin >> n;
        cout << "4\n0 0\n1 0\n1 1\n0 1\n";
        vector<pii> ans = {
                {1, 0},
                {1, 1},
                {0, 1},
                {-1, 1},
                {-1, 0},
                {-1, -1},
                {0, -1},
                {1, -1},
        };
        FOR(i, 0, n)
                cout << ans[i] << '\n';
}

int32_t main() {
        cin.tie(0);
        ios_base::sync_with_stdio(0);
        int t = 1;
        //freopen("in.txt", "r", stdin);
        //freopen("out.txt", "w", stdout);
        //cin >> t;
        while (t--)
                solve();
}
```