

# Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	20	25	445

## Task A ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

void solve() {
    int k;
    cin >> k;
    if (k == 1)
        cout << 1;
    else
        cout << "098765432"[(9 - (k - 1) % 9) % 9];
}

int main() {
    srand(chrono::duration_cast<chrono::nanoseconds>(chrono::high_resolution_clock::now().
        time_since_epoch()).count());
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int t = 1;
    //cin >> t;
    while (t--> 0) {
        solve();
    }

    return 0;
}
```

## Task B ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

int cur_letters[30];

void solve() {
    int n, k;
    string s;
    cin >> n >> k >> s;

    int cur_blocks = 0;

    queue <pair <char, int>> q;

    ++cur_letters[s[0] - 'a'];
    int diff_letters = 1;

    q.push({s[0], 0});
    for (int i = 1; i < n; ++i) {
        if (i - q.front().second + 1 > k) {
            //cout << i - 1 << " block end\n";
            ++cur_blocks;

            for (int j = 0; j < 30; ++j)
                cur_letters[j] = 0;
            diff_letters = 0;

            q = queue <pair <char, int>>();
        }

        if (!(diff_letters < 3 || cur_letters[s[i] - 'a'])) {
            //cout << i - 1 << " block end\n";
            ++cur_blocks;

            for (int j = 0; j < 30; ++j)
                cur_letters[j] = 0;
            diff_letters = 0;

            q = queue <pair <char, int>>();
        }

        ++cur_letters[s[i] - 'a'];
        if (cur_letters[s[i] - 'a'] == 1)
            ++diff_letters;
        q.push({s[i], i});
    }
    cout << cur_blocks + (q.empty() ? 0 : 1);
}

int main() {
    srand(chrono::duration_cast<chrono::nanoseconds>(chrono::high_resolution_clock::now().
        time_since_epoch()).count());
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int t = 1;
    //cin >> t;
    while (t--) {
        solve();
    }

    return 0;
}
```

## Task C ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

const int INF = 1e9;
int v[501], w[501], i;
int min_w[2][250001];
char par[501][250001]; // 0 - += v, 1 - += w

void solve() {
    int n, x, y;
    cin >> n >> x >> y;
    for (i = n - 1; i >= 0; --i)
        cin >> v[i];
    for (i = n - 1; i >= 0; --i)
        cin >> w[i];
    for (i = 0; i <= x; ++i)
        min_w[0][i] = INF;
    min_w[0][v[0]] = 0;
    min_w[0][0] = w[0];
    par[0][v[0]] = 0;
    par[0][0] = 1;

    for (i = 1; i < n; ++i) {
        for (int v_sm = 0; v_sm <= x; ++v_sm) {
            min_w[i % 2][v_sm] = INF;
            if (min_w[1 - i % 2][v_sm] != INF) {
                min_w[i % 2][v_sm] = min_w[1 - i % 2][v_sm] + w[i];
                par[i][v_sm] = 1;
            }
            if (v_sm >= v[i] && min_w[i % 2][v_sm] > min_w[1 - i % 2][v_sm - v[i]]) {
                min_w[i % 2][v_sm] = min_w[1 - i % 2][v_sm - v[i]];
                par[i][v_sm] = 0;
            }
        }
    }

    int res_v = -1;
    for (i = 0; i <= x; ++i) {
        if (min_w[(n - 1) % 2][i] <= y) {
            res_v = i;
            break;
        }
    }
    //cout << res_v << '\n';
    if (res_v == -1) {
        cout << -1;
        return;
    }
    int cur_v = res_v;
    for (i = n - 1; i >= 0; --i) {
        cout << (par[i][cur_v] ? 'y' : 'x');
        if (!par[i][cur_v])
            cur_v -= v[i];
    }
}

int main() {
    srand(chrono::duration_cast<chrono::nanoseconds>(chrono::high_resolution_clock::now().
        time_since_epoch()).count());
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int t = 1;
    //cin >> t;
```

```
    while (t--) {  
        solve();  
    }  
  
    return 0;  
}
```

## Task D ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

const int MAX_N = 1e5 + 1, INF = 1e9;
int dp[MAX_N];

int type(char c) {
    return c == '[' || c == ']';
}

void solve() {
    int n;
    string s;

    cin >> n >> s;
    n *= 2;
    vector<int> confl;
    for (int i = 0; i < n; ++i) {
        int q = type(s[i]);
        if (confl.empty() || confl.back() != q)
            confl.push_back(q);
        else
            confl.pop_back();
    }
    cout << confl.size() / 2;
}

int main() {
    srand(chrono::duration_cast<chrono::nanoseconds>(chrono::high_resolution_clock::now().
        time_since_epoch()).count());
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int t = 1;
    //cin >> t;
    while (t--) {
        solve();
    }

    return 0;
}
```

## Task E ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

map <tuple<int , int , int>, int> answ;

int build(int n, vector <int> & pick) {
    sort(pick.begin(), pick.end());
    auto it = answ.find({pick[0], pick[1], pick[2]});
    assert(it != answ.end());
    return it->second;
}

int solve(int n, vector <int> & pick) {
    sort(pick.begin(), pick.end());
    vector <int> answs;
    do {
        int ans = answ[{pick[0], pick[1], pick[2]}];
        if (ans == pick[3])
            answs.push_back(pick[3]);
    } while (next_permutation(pick.begin(), pick.end()));
    sort(answs.begin(), answs.end());
    return answs.back();
}

void add() {
    int n, k;
    cin >> n >> k;
    if (n == 1000000) {
        cout << 911911;
    } else if (n == 10 && k == 3) {
        vector <int> pick(k);
        for (auto & e: pick)
            cin >> e;
        cout << build(n, pick);
    }
    cout << '\n';
}

void clear() {
    int n, k;
    cin >> n >> k;
    if (n == 1000000) {
        for (int i = 0; i <= k; ++i) {
            int numb;
            cin >> numb;
            if (numb != 911911)
                cout << numb << '␣';
        }
    } else if (n == 10 && k == 3) {
        vector <int> pick(k + 1);
        for (auto & e : pick)
            cin >> e;
        int ans = solve(n, pick);
        for (auto e : pick)
            if (e != ans)
                cout << e << '␣';
    }
    cout << '\n';
}

int main() {
    ios_base::sync_with_stdio(false);
```

```

cin.tie(nullptr);

srand(15);
for (int i = 1; i <= 10; ++i) {
    for (int j = i + 1; j <= 10; ++j) {
        for (int k = j + 1; k <= 10; ++k) {
            int r = 1 + rand() % 10;
            while (i == r || j == r || k == r)
                r = 1 + rand() % 10;
            answ[{i, j, k}] = r;
        }
    }
}
string type;
int t;
cin >> type >> t;
if (type == "add") {
    while (t--)
        add();
} else {
    while (t--)
        clear();
}

return 0;
}

```

## Task F ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

int sh_x[] = {-1, -1, -1, 0, 0, 1, 1, 1};
int sh_y[] = {1, 0, -1, -1, 1, 1, 0, -1};

void solve() {
    int n;
    cin >> n;
    assert(n <= 8);
    cout << "4\n0_0\n1_0\n1_1\n0_1\n";
    for (int i = 0; i < n; ++i)
        cout << sh_x[i] << '_' << sh_y[i] << '\n';
}

int main() {
    srand(chrono::duration_cast<chrono::nanoseconds>(chrono::high_resolution_clock::now().
        time_since_epoch()).count());
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int t = 1;
    //cin >> t;
    while (t--) {
        solve();
    }

    return 0;
}
```