

# 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>
#include <immintrin.h>

using namespace std;

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
// #pragma GCC optimize("O3")
// #pragma GCC optimize("unroll-loops")

template<typename T>
istream &operator>>(istream &in, vector<T> &a) {
    for (auto &i : a)
        in >> i;
    return in;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &a) {
    for (auto &i : a) {
        out << i << "_";
    }
    return out;
}

template<typename T, typename D>
istream &operator>>(istream &in, pair<T, D> &a) {
    in >> a.first >> a.second;
    return in;
}

template<typename T, typename D>
ostream &operator<<(ostream &out, const pair<T, D> &a) {
    out << a.first << "_" << a.second;
    return out;
}

struct LogOutput {
    template<typename T>
    LogOutput &operator<<(const T &x) {
#ifdef DIVAN
        cout << x;
#endif
        return *this;
    }
} dout, fout;

typedef long long ll;
typedef unsigned long long ull;
typedef double dl;
typedef complex<double> cd;

#define nl '\n'
#define elif else if
#define all(_v) _v.begin(), _v.end()
#define rall(v) v.rbegin(), v.rend()
```

```

#define sz(v) (int)(v.size())
#define sqr(_v) ((_v) * (_v))
#define vpi vector<pair<int, int>>
#define eb emplace_back
#define pb push_back
#define mod(x, m) ((x) >= 0 ? ((x) % m) : (((x) % m) + m) % m)
#define vi vector<int>
#define pi pair<int, int>
#define ti tuple<int, int, int>
#define minq(x, y) x = min((x), (y))
#define maxq(x, y) x = max(x, (y))
#define forn(i, n) for (int i = 0; i < (n); ++i)

const ll INFL = 9187201950435737471;
const ll nINFL = -9187201950435737472;
const int INF = 2139062143;
const int nINF = -2139062144;
const ull ULINF = numeric_limits<ull>::max();
const long double PI = acos(-1);
auto seed = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(seed);

inline void IO() {
#ifdef DIVAN
    freopen("../input.txt", "r", stdin);
    freopen("../output.txt", "w", stdout);
#else
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
#endif
}

void Solve() {
    int n;
    cin >> n;
    if (n <= 10) {
        cout << n % 10;
    } else {
        n -= 11;
        n %= 9;
        int x = n + 2;
        x %= 10;
        cout << x;
    }
}

signed main() {
    IO();
    int t = 1;
    //     cin >> t;
    int startTime = clock();
    for (int i = 1; i <= t; ++i) {
        //         cout << "Case #" << i << ": ";
        Solve();
    }
    int endTime = clock();
    fout << '\n' << "Time:_" << (endTime - startTime + 999) / 1000;
    return 0;
}

```

## Task B ()

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

using namespace std;

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
// #pragma GCC optimize("O3")
// #pragma GCC optimize("unroll-loops")

template<typename T>
istream &operator>>(istream &in, vector<T> &a) {
    for (auto &i : a)
        in >> i;
    return in;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &a) {
    for (auto &i : a) {
        out << i << "_";
    }
    return out;
}

template<typename T, typename D>
istream &operator>>(istream &in, pair<T, D> &a) {
    in >> a.first >> a.second;
    return in;
}

template<typename T, typename D>
ostream &operator<<(ostream &out, const pair<T, D> &a) {
    out << a.first << "_ " << a.second;
    return out;
}

struct LogOutput {
    template<typename T>
    LogOutput &operator<<(const T &x) {
#ifdef DIVAN
        cout << x;
#endif
        return *this;
    }
} dout, fout;

typedef long long ll;
typedef unsigned long long ull;
typedef double dl;
typedef complex<double> cd;

#define nl '\n'
#define elif else if
#define all(_v) _v.begin(), _v.end()
#define rall(v) v.rbegin(), v.rend()
#define sz(v) (int)(v.size())
#define sqr(_v) ((_v) * (_v))
#define vpi vector<pair<int, int>>
#define eb emplace_back
#define pb push_back
#define mod(x, m) ((x) >= 0 ? ((x) % m) : (((x) % m) + m) % m)
#define vi vector<int>
#define pi pair<int, int>
#define ti tuple<int, int, int>
#define minq(x, y) x = min((x), (y))
#define maxq(x, y) x = max(x, (y))
#define forn(i, n) for (int i = 0; i < (n); ++i)

const ll INFL = 9187201950435737471;
const ll nINFL = -9187201950435737472;
const int INF = 2139062143;
```

```

const int nINF = -2139062144;
const ull ULINF = numeric_limits<ull>::max();
const long double PI = acos(-1);
auto seed = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(seed);

inline void IO() {
#ifdef DIVAN
    freopen("../input.txt", "r", stdin);
    freopen("../output.txt", "w", stdout);
#else
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
#endif
}

const int MAXN = 26;

void Solve() {
    int n, k;
    cin >> n >> k;
    string s;
    cin >> s;
    int ans = 0;
    int last = 0;
    map<int, int> cnt;
    forn(i, n) {
        cnt[s[i]]++;
        if (sz(cnt) > 3 || i - last + 1 > k) {
            ++ans;
            cnt.clear();
            last = i;
        }
        cnt[s[i]] = 1;
    }
    cout << ans + 1;
}

signed main() {
    IO();
    int t = 1;
    //     cin >> t;
    int startTime = clock();
    for (int i = 1; i <= t; ++i) {
        //         cout << "Case #" << i << ": ";
        Solve();
    }
    int endTime = clock();
    fout << '\n' << "Time:_" << (endTime - startTime + 999) / 1000;
    return 0;
}

```

## Task C ()

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

using namespace std;

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
// #pragma GCC optimize("O3")
// #pragma GCC optimize("unroll-loops")

template<typename T>
istream &operator>>(istream &in, vector<T> &a) {
    for (auto &i : a)
        in >> i;
    return in;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &a) {
    for (auto &i : a) {
        out << i << "_";
    }
    return out;
}

template<typename T, typename D>
istream &operator>>(istream &in, pair<T, D> &a) {
    in >> a.first >> a.second;
    return in;
}

template<typename T, typename D>
ostream &operator<<(ostream &out, const pair<T, D> &a) {
    out << a.first << "_" << a.second;
    return out;
}

struct LogOutput {
    template<typename T>
    LogOutput &operator<<(const T &x) {
#ifdef DIVAN
        cout << x;
#endif
        return *this;
    }
} dout, fout;

typedef long long ll;
typedef unsigned long long ull;
typedef double dl;
typedef complex<double> cd;

#define nl '\n'
#define elif else if
#define all(_v) _v.begin(), _v.end()
#define rall(v) v.rbegin(), v.rend()
#define sz(v) (int)(v.size())
#define sqr(_v) ((_v) * (_v))
#define vpi vector<pair<int, int>>
#define eb emplace_back
#define pb push_back
#define mod(x, m) ((x) >= 0 ? ((x) % m) : (((x) % m) + m) % m)
#define vi vector<int>
#define pi pair<int, int>
#define ti tuple<int, int, int>
#define minq(x, y) x = min((x), (y))
#define maxq(x, y) x = max(x, (y))
#define forn(i, n) for (int i = 0; i < (n); ++i)

const ll INFL = 9187201950435737471;
const ll nINFL = -9187201950435737472;
```

```

const int INF = 2139062143;
const int nINF = -2139062144;
const ull ULINF = numeric_limits<ull>::max();
const long double PI = acos(-1);
auto seed = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(seed);

inline void IO() {
#ifdef DIVAN
    freopen("../input.txt", "r", stdin);
    freopen("../output.txt", "w", stdout);
#else
    //    freopen("input.txt", "r", stdin);
    //    freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
#endif
}

const int MAXN = 250000 + 15;
const int MAXE = 500 + 15;

int dp[MAXN];
int p[MAXE][MAXN];

void Solve() {
    memset(dp, -1, sizeof(dp));
    memset(p, -1, sizeof(p));
    int n, x, y;
    cin >> n >> x >> y;
    vector<int> v(n), w(n);
    cin >> v >> w;
    dp[0] = 0;
    forn(i, n) {
        for (int now = x; now >= 0; --now) {
            if (i != 0) {
                p[i][now] = p[i - 1][now];
            }
            if (dp[now] == -1 || now + v[i] > x) {continue;}
            if (dp[now + v[i]] < dp[now] + w[i]) {
                dp[now + v[i]] = dp[now] + w[i];
                p[i][now + v[i]] = i;
            }
        }
    }
    int mx = x;
    for (int i = x; i >= 0; --i) {
        if (dp[i] > dp[mx]) {
            mx = i;
        }
    }
    int sum2 = 0;
    for (auto u : w) {
        sum2 += u;
    }
    if (sum2 - dp[mx] > y) {
        cout << "-1";
        return;
    }
    vector<char> ans(n, 'y');
    int me = mx;
    int i = n - 1;
    while(i >= 0) {
        int x = p[i][me];
        if (x != i) {
            --i;
            continue;
        }
        ans[p[i][me]] = 'x';
        me -= v[p[i][me]];
        --i;
    }
    for (auto u : ans) {
        cout << u;
    }
}

```

```

    }
}

signed main() {
    IO();
    int t = 1;
    //    cin >> t;
    int startTime = clock();
    for (int i = 1; i <= t; ++i) {
        //    cout << "Case #" << i << ": ";
        Solve();
    }
    int endTime = clock();
    fout << '\n' << "Time:_" << (endTime - startTime + 999) / 1000;
    return 0;
}

```

## Task D ()

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

using namespace std;

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
// #pragma GCC optimize("O3")
// #pragma GCC optimize("unroll-loops")

template<typename T>
istream &operator>>(istream &in, vector<T> &a) {
    for (auto &i : a)
        in >> i;
    return in;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &a) {
    for (auto &i : a) {
        out << i << "_";
    }
    return out;
}

template<typename T, typename D>
istream &operator>>(istream &in, pair<T, D> &a) {
    in >> a.first >> a.second;
    return in;
}

template<typename T, typename D>
ostream &operator<<(ostream &out, const pair<T, D> &a) {
    out << a.first << "_" << a.second;
    return out;
}

struct LogOutput {
    template<typename T>
    LogOutput &operator<<(const T &x) {
#ifdef DIVAN
        cout << x;
#endif
        return *this;
    }
} dout, fout;

typedef long long ll;
typedef unsigned long long ull;
typedef double dl;
typedef complex<double> cd;

#define nl '\n'
#define elif else if
#define all(_v) _v.begin(), _v.end()
#define rall(v) v.rbegin(), v.rend()
#define sz(v) (int)(v.size())
#define sqr(_v) ((_v) * (_v))
#define vpi vector<pair<int, int>>
#define eb emplace_back
#define pb push_back
#define mod(x, m) ((x) >= 0 ? ((x) % m) : (((x) % m) + m) % m)
#define vi vector<int>
#define pi pair<int, int>
#define ti tuple<int, int, int>
#define minq(x, y) x = min((x), (y))
#define maxq(x, y) x = max(x, (y))
#define forn(i, n) for (int i = 0; i < (n); ++i)

const ll INFL = 9187201950435737471;
const ll nINFL = -9187201950435737472;
```



```

const int INF = 2139062143;
const int nINF = -2139062144;
const ull ULINF = numeric_limits<ull>::max();
const long double PI = acos(-1);
auto seed = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(seed);

inline void IO() {
#ifdef DIVAN
    freopen("../input.txt", "r", stdin);
    freopen("../output.txt", "w", stdout);
#else
    //    freopen("input.txt", "r", stdin);
    //    freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
#endif
}

void Solve() {
    int n;
    cin >> n;
    n *= 2;
    stack<char> q;
    forn(i, n) {
        char x;
        cin >> x;
        if (x == ')' || x == '(') {
            x = '(';
        } else {
            x = '[';
        }
        if (!q.empty() && q.top() == x) {
            q.pop();
        } else {
            q.push(x);
        }
    }
    cout << sz(q) / 2;
}

signed main() {
    IO();
    int t = 1;
    //    cin >> t;
    int startTime = clock();
    for (int i = 1; i <= t; ++i) {
        //        cout << "Case #" << i << ": ";
        Solve();
    }
    int endTime = clock();
    fout << '\n' << "Time:_" << (endTime - startTime + 999) / 1000;
    return 0;
}

```

## Task E ()

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

using namespace std;

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
// #pragma GCC optimize("O3")
// #pragma GCC optimize("unroll-loops")

template<typename T>
istream &operator>>(istream &in, vector<T> &a) {
    for (auto &i : a)
        in >> i;
    return in;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &a) {
    for (auto &i : a) {
        out << i << "_";
    }
    return out;
}

template<typename T, typename D>
istream &operator>>(istream &in, pair<T, D> &a) {
    in >> a.first >> a.second;
    return in;
}

template<typename T, typename D>
ostream &operator<<(ostream &out, const pair<T, D> &a) {
    out << a.first << "_" << a.second;
    return out;
}

struct LogOutput {
    template<typename T>
    LogOutput &operator<<(const T &x) {
#ifdef DIVAN
        cout << x;
#endif
        return *this;
    }
} dout, fout;

typedef long long ll;
typedef unsigned long long ull;
typedef double dl;
typedef complex<double> cd;

#define nl '\n'
#define elif else if
#define all(_v) _v.begin(), _v.end()
#define rall(v) v.rbegin(), v.rend()
#define sz(v) (int)(v.size())
#define sqr(_v) ((_v) * (_v))
#define vpi vector<pair<int, int>>
#define eb emplace_back
#define pb push_back
#define mod(x, m) ((x) >= 0 ? ((x) % m) : (((x) % m) + m) % m)
#define vi vector<int>
#define pi pair<int, int>
#define ti tuple<int, int, int>
#define minq(x, y) x = min((x), (y))
#define maxq(x, y) x = max(x, (y))
#define forn(i, n) for (int i = 0; i < (n); ++i)

const ll INFL = 9187201950435737471;
const ll nINFL = -9187201950435737472;
```

```

const int INF = 2139062143;
const int nINF = -2139062144;
const ull ULINF = numeric_limits<ull>::max();
const long double PI = acos(-1);
//auto seed = chrono::high_resolution_clock::now().time_since_epoch().count();
auto seed = 0;
mt19937 rnd(seed);

inline void IO() {
#ifdef DIVAN
    freopen("../input.txt", "r", stdin);
    freopen("../output.txt", "w", stdout);
#else
    //    freopen("input.txt", "r", stdin);
    //    freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
#endif
}

void Solve() {
    int n, k;
    cin >> n >> k;
    vi used(n + 1, 0);
    forn(i, k) {
        int x;
        cin >> x;
        used[x] = 1;
    }
    for (int i = 1; i + 2 <= n; ++i) {
        if (!used[i] && !used[i + 1] && !used[i + 2]) {
            cout << i << '\n';
            return;
        }
    }
    assert(false);
}

void Solve2() {
    int n, k;
    cin >> n >> k;
    vector<bool> used(n + 1, 0);
    forn(i, k + 1) {
        int x;
        cin >> x;
        used[x] = 1;
    }
    for (int i = 1; i + 2 <= n; ++i) {
        if (used[i] && !used[i + 1] && !used[i + 2]) {
            forn(j, n + 1) {
                if (used[j] && j != i) {
                    cout << j << '\u';
                }
            }
            return;
        }
    }
    assert(false);
}

signed main() {
    IO();
    string s;
    cin >> s;
    int t = 1;
    cin >> t;
    int startTime = clock();
    for (int i = 1; i <= t; ++i) {
        //    cout << "Case #" << i << ": ";
        if (s == "add") {
            Solve();
        } else {
            Solve2();
        }
    }
}

```

```
}  
int endTime = clock();  
fout << '\n' << "Time:␣" << (endTime - startTime + 999) / 1000;  
return 0;  
}
```

## Task F ()

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

using namespace std;

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
// #pragma GCC optimize("O3")
// #pragma GCC optimize("unroll-loops")

template<typename T>
istream &operator>>(istream &in, vector<T> &a) {
    for (auto &i : a)
        in >> i;
    return in;
}

template<typename T>
ostream &operator<<(ostream &out, const vector<T> &a) {
    for (auto &i : a) {
        out << i << "_";
    }
    return out;
}

template<typename T, typename D>
istream &operator>>(istream &in, pair<T, D> &a) {
    in >> a.first >> a.second;
    return in;
}

template<typename T, typename D>
ostream &operator<<(ostream &out, const pair<T, D> &a) {
    out << a.first << "_" << a.second;
    return out;
}

struct LogOutput {
    template<typename T>
    LogOutput &operator<<(const T &x) {
#ifdef DIVAN
        cout << x;
#endif
        return *this;
    }
} dout, fout;

typedef long long ll;
typedef unsigned long long ull;
typedef double dl;
typedef complex<double> cd;

#define nl '\n'
#define elif else if
#define all(_v) _v.begin(), _v.end()
#define rall(v) v.rbegin(), v.rend()
#define sz(v) (int)(v.size())
#define sqr(_v) ((_v) * (_v))
#define vpi vector<pair<int, int>>
#define eb emplace_back
#define pb push_back
#define mod(x, m) ((x) >= 0 ? ((x) % m) : (((x) % m) + m) % m)
#define vi vector<int>
#define pi pair<int, int>
#define ti tuple<int, int, int>
#define minq(x, y) x = min((x), (y))
#define maxq(x, y) x = max(x, (y))
#define forn(i, n) for (int i = 0; i < (n); ++i)

const ll INFL = 9187201950435737471;
const ll nINFL = -9187201950435737472;
```

```

const int INF = 2139062143;
const int nINF = -2139062144;
const ull ULINF = numeric_limits<ull>::max();
const long double PI = acos(-1);
auto seed = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(seed);

inline void IO() {
#ifdef DIVAN
    freopen("../input.txt", "r", stdin);
    freopen("../output.txt", "w", stdout);
#else
    //    freopen("input.txt", "r", stdin);
    //    freopen("output.txt", "w", stdout);
    ios_base::sync_with_stdio(0);
    cin.tie(0);
#endif
}

void Solve() {
    int n;
    cin >> n;
    cout << "4\n";
    cout << "0_0\n" << "0_1\n" << "1_1\n" << "1_0\n";
    vpi vec = {{-1, -1}, {-1, 0}, {-1, 1}, {0, -1}, {0, 1}, {1, -1}, {1, 0}, {1, 1}};
    assert(n <= sz(vec));
    forn(i, n) {
        cout << vec[i] << '\n';
    }
}

signed main() {
    IO();
    //    string s;
    //    cin >> s;
    int t = 1;
    //    cin >> t;
    int startTime = clock();
    for (int i = 1; i <= t; ++i) {
        cout << "Case #" << i << ": ";
        //        if (s == "add") {
        //            Solve();
        //        } else {
        //            Solve2();
        //        }
    }
    int endTime = clock();
    fout << '\n' << "Time:_" << (endTime - startTime + 999) / 1000;
    return 0;
}

```