Олимпиада СПбГУ по информатике 2020/21 учебного года

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

Task A ()

```
#pragma GCC optimize("O3")
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <utility>
#include <algorithm>
#include <cassert>
#include <climits>
#define int long long
#define all(z) z.begin(), z.end()
#define rall(z) z.rbegin(), z.rend()
#define pb push_back
#define eb emplace_back
#define int long long
#define u64 uint64_t
using namespace std;
using vi = vector<int>;
using vc = vector<char>;
using pii = pair<int, int>;
using vvi = vector<vector<int>>;


signed main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int k;
    cin >> k;

    k--;
    if (k < 9)
    {
        cout << k+1;
    }
    else
    {
        k %= 9;
        if (k == 0)
            cout << "0";
        else
            cout << k+1;
    }

    return 0;
}
```

Task B ()

```
#pragma GCC optimize("O3")
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <utility>
#include <algorithm>
#include <cassert>
#include <climits>
#define int long long
#define all(z) z.begin(), z.end()
#define rall(z) z.rbegin(), z.rend()
#define pb push_back
#define eb emplace_back
#define int long long
#define u64 uint64_t
using namespace std;
using vi = vector<int>;
using vc = vector<char>;
using pii = pair<int, int>;
using vvi = vector<vector<int>>;


signed main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n, k;
    string s;
    cin >> n >> k >> s;

    int ans = 1;
    set<char> cs;
    int len = 0;
    for (int i = 0; i < n; i++)
    {
        cs.insert(s[i]);
        len++;

        if (cs.size() > 3 || len > k)
        {
            cs = {s[i]};
            len = 1;
            ans++;
        }
    }

    cout << ans;

    return 0;
}
```

Task C ()

```
#pragma GCC optimize("O3")
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <utility>
#include <algorithm>
#include <cassert>
#include <climits>
#include <cstring>
#define int long long
#define all(z) z.begin(), z.end()
#define rall(z) z.rbegin(), z.rend()
#define pb push_back
#define eb emplace_back
#define u64 uint64_t
using namespace std;
using vi = vector<int>;
using vc = vector<char>;
using pii = pair<int, int>;
using vvi = vector<vector<int>>;


template <typename T1, typename T2>
inline void rmax(T1 &a, T1 b, T2 &c, T2 d)
{
    if (a < b)
    {
        a = b;
        c = d;
    }
}


constexpr int N{500}, S{250000};
int a[N], b[N];
//int dp[N+1][S+1];
char from[N+1][S+1]; // 0 - bag_x, 1 - bag_y, step_back


signed main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n, x, y;
    cin >> n >> x >> y;

    for (int i = 0; i < n; i++)
        cin >> a[i];
    for (int i = 0; i < n; i++)
        cin >> b[i];

    vi dpl(x+1, -1), dp;
    dpl[0] = y;

    for (int i = 1; i <= n; i++)
    {
        dp.assign(x+1, -1);

        for (int x0 = 0; x0 <= x; x0++)
        {
            if (x0 > 0)
                rmax(dp[x0], dp[x0-1],
                     from[i][x0], (char)2);

            int wx = a[i-1], wy = b[i-1];
            if (wx <= x0)
                rmax(dp[x0], dpl[x0-wx],
```

```cpp
                     from[i][x0], (char)0);
            if (wy <= dpl[x0])
                rmax(dp[x0], dpl[x0] - wy,
                     from[i][x0], (char)1);
        }

        swap(dp, dpl);
    }

    if (dpl[x] < 0)
        cout << "-1";
    else
    {
        vc ans(n);
        int i = n, j = x;
        while (i > 0)
        {
            //cout << i << ' ' << j << ": " << (int)from[i][j] << endl;
            if (from[i][j] == 2)
                j--;
            else if (from[i][j] == 0)
            {
                ans[i-1] = 0;
                j -= a[i-1];
                i--;
            }
            else if (from[i][j] == 1)
            {
                ans[i-1] = 1;
                i--;
            }
        }

        for (int i = 0; i < n; i++)
            cout << (ans[i] ? 'y' : 'x');
    }

    return 0;
}

/*
1 10 10
10
250000
}*/
```

Task D ()

```
#pragma GCC optimize("O3")
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <utility>
#include <algorithm>
#include <cassert>
#include <climits>
#include <cstring>
#include <stack>
#define all(z) z.begin(), z.end()
#define rall(z) z.rbegin(), z.rend()
#define pb push_back
#define eb emplace_back
using namespace std;
using vi = vector<int>;
using vc = vector<char>;
using pii = pair<int, int>;
using vvi = vector<vector<int>>;


inline int strip(const vi &v)
{
    stack<int> s;
    for (int x : v)
        if (s.empty() || s.top() != x)
            s.push(x);
        else
            s.pop();
    return s.size();
}


signed main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n;
    cin >> n;

    vi a(2*n, 0);
    for (int i = 0; i < 2*n; i++)
    {
        char c;
        cin >> c;
        if (c == '[' || c == ']')
            a[i] = 1;
    }

    cout << strip(a) / 2;

    return 0;
}

/*** HELPER CODE
for (int n = 2; n <= 100; n += 2)
{
    vi s(n);
    for (int i = 0; i < n; i++)
        s[i] = i % 2;

    int c = n;
    for (int mask = 0; mask < 1ll<<n; mask++)
    {
        for (int i = 0; i < n; i++)
            if (mask >> i & 1)
                s[i] ^= 1;
        if (ok(s))
        {
            //cout << mask << '\n';
            c = min(c, __builtin_popcount(mask));
        }
        for (int i = 0; i < n; i++)
            if (mask >> i & 1)
                s[i] ^= 1;
    }

    cout << n << ": " << c << endl;
}
***/
```

Task E ()

```
#pragma GCC optimize("O3")
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <utility>
#include <algorithm>
#include <cassert>
#include <climits>
#include <cstring>
#include <stack>
#include<random>
#define int long long
#define all(z) z.begin(), z.end()
#define rall(z) z.rbegin(), z.rend()
#define pb push_back
#define eb emplace_back
using namespace std;
using vi = vector<int>;
using vc = vector<char>;
using pii = pair<int, int>;
using vvi = vector<vector<int>>;
mt19937 rng(3);


inline int binpow(int x, int n, int mod)
{
    int res = 1;
    while (n)
    {
        if (n & 1)
            res = res*x % mod;
        x = x*x % mod;
        n >>= 1;
    }
    return res;
}

constexpr int p{110017};
constexpr int N{1000000};
int ch[N];
int n, k;

inline int find_place(const set<int> &s, int x)
{
    assert(s.size() < n);

    while (true)
    {
        int y = ch[x];
        auto it = s.lower_bound(y);
        if (it == s.end() || *it != y)
            break;
        x = (x+1) % n;
    }

    return ch[x];
}

inline int is_scan(vi a, int n)
{
    if (a.empty())
        return 0;

    sort(all(a));
    vc c(n, 0);
    int d = 0;
    for (int x : a)
    {
        int l = (x+n-1)%n;
        int r = (x+1)%n;
        if (!c[l] && !c[r])
            d++;
        c[x] = true;
    }

    if (d > 1)
        return -1;

    int r = 0;
    while (!c[r])
        r++;
    while (c[r])
        r = (r+1)%n;
    return r;
}

inline int ex_scan(vi a, int n)
{
    sort(all(a));
    vc c(n, 0);

    for (int x : a)
        c[x] = true;

    int r = 0;
    while (!c[r])
        r++;
    while (c[r])
        r=(r+1)%n;
    r = (r+n-1) % n;
    return r;
}

inline void encode()
{
    vi a(k);
    for (int i = 0; i < k; i++)
    {
        cin >> a[i];
        a[i]--;
    }

    if (n == 10 && k == 3)
    {
        vi a1, a2;
        for (int i = 0; i < k; i++)
            if (a[i] < 5)
                a1.pb(a[i]);
            else
                a2.pb(a[i] - 5);

        int is1 = is_scan(a1, 5);
        int is2 = is_scan(a2, 5);

        int ans = -1;
        if (is1 != -1)
        {
            ans = is1;
        }
        else
        {
            assert(is2 != -1);
            ans = 5 + is2;
        }

        cout << ans+1 << '\n';
        return;
    }

    sort(all(a));
    set<int> sa(all(a));
    int pos = 0;
    for (int i = 0; i < k; i++)
        pos = (pos + binpow(p, a[i], n)) % n;

    cout << find_place(sa, pos) + 1 << '\n';
}

inline void decode()
{
    vi a(k+1);
    for (int i = 0; i < k+1; i++)
    {
        cin >> a[i];
        a[i]--;
    }

    if (n == 10 && k == 3)
    {
        vi a1, a2;
        for (int i = 0; i < k+1; i++)
            if (a[i] < 5)
                a1.pb(a[i]);
            else
                a2.pb(a[i] - 5);

        int is1 = is_scan(a1, 5);
        int is2 = is_scan(a2, 5);

        int ans = -1;
        if (is1 != -1)
        {
            ans = ex_scan(a1, 5);
        }
        else
        {
            assert(is2 != -1);
            ans = 5 + ex_scan(a2, 5);
        }

        //cout << "A " << ans << '\n';
        for (int x : a)
            if (x != ans)
                cout << x+1 << '␣';
        cout << '\n';
        return;
    }

    sort(all(a));
    set<int> sa(all(a));

    vi b(k+1);
    int sum = 0;
    for (int i = 0; i < k+1; i++)
    {
        b[i] = binpow(p, a[i], n);
        sum = (sum+b[i]) % n;
    }

    for (int i = 0; i < k+1; i++)
    {
        sa.erase(a[i]);
        sum = (sum-b[i]+n) % n;

        if (find_place(sa, sum) == a[i])
        {
            for (int j = 0; j < k+1; j++)
                if (j != i)
                    cout << a[j]+1 << '␣';
            cout << '\n';
            return;
        }

        sum = (sum+b[i]) % n;
        sa.insert(a[i]);
    }

    //assert(false); // FAIL
}


signed main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    string s;
    cin >> s;

    int t;
    cin >> t >> n >> k;

    for (int i = 0; i < n; i++)
        ch[i] = i;
    shuffle(ch, ch+n, rng);

    if (s == "add")
        while (t--)
            encode();
    else if (s == "clear")
        while (t--)
            decode();

    return 0;
}


/*** HELPER CODE
for (int p = 110000; true; p++)
{
    bool ok = true;
    for (int d = 2; d*d <= p && ok; d++)
    {
        if (p % d == 0)
            ok = false;
    }
    if (ok)
    {
        cout << p;
        return 0;
    }
}
***/
```

Task F ()

```
#pragma GCC optimize("O3")
#include <iostream>
#include <iomanip>
#include <cstdio>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <utility>
#include <algorithm>
#include <cassert>
#include <climits>
#include <cstring>
#define int long long
#define all(z) z.begin(), z.end()
#define rall(z) z.rbegin(), z.rend()
#define pb push_back
#define eb emplace_back
using namespace std;
using vi = vector<int>;
using vc = vector<char>;
using pii = pair<int, int>;
using vvi = vector<vector<int>>;


inline void put(int x, int y)
{
    cout << x << '␣' << y << '\n';
}


signed main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n;
    cin >> n;

    cout << "4\n";
    put(0, 0);
    put(1, 0);
    put(1, 1);
    put(0, 1);

    vector<pii> off = {
        {1, -1},
        {1, 0},
        {1, 1},
        {0, 1},
        {-1, 1},
        {-1, 0},
        {-1, -1},
        {0, -1},
    };
    assert(n <= 8);

    for (int i = 0; i < n; i++)
        put(off[i].first, off[i].second);

    return 0;
}
```