Олимпиада СПбГУ по информатике 2020/21 учебного года

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

Task A ()

```cpp
#pragma GCC optimize("Ofast")
#include <iostream>
#include <algorithm>
#include <string.h>
#include <cstdlib>
#include <vector>

#define ll long long
using namespace std;

int main() {
    std::ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n;
    cin>>n;
    ll ans=0;
    ll add=0;
    for(ll i=1;i<=n;++i)
    {
        ans=i+add;
        add=ans/10;
    }
    cout<<ans%10;
    return 0;
}
```

Task B ()

```
#pragma GCC optimize("Ofast")
#include <iostream>
#include <algorithm>
#include <string.h>
#include <cstdlib>
#include <map>
#include<cmath>
#include <vector>
#include <set>
#include <cstdlib>
#include <string.h>
#include <unordered_set>
#define ll long long

using namespace std;

int dp[100000][26];
int get(int l, int r) {
    int kol = 0;
    for (int j = 0; j < 26; ++j) {
        int a = dp[r][j];
        if (l)
            a -= dp[l-1][j];
        if (a)
            kol++;
    }
    return kol;
}
bool ok(int i, int m) {
    return get(i - m + 1, i) <= 3;
}
struct node {
    int val, l, r;

    node() : val(1e9), l(-1), r(-1) {}

    node(int val, int l, int r) : val(val), l(l), r(r) {}

    node operator+(const node &Node) const {
        return node(min(val, Node.val), l, Node.r);
    }
};
const int N=1<<17;
node tree[2*N];
void build()
{
    for(int i=0;i<N;++i)
        tree[i+N]=node(1e9, i, i);
    for(int i=N-1;i>=1;--i)
    {
        tree[i]=tree[i*2]+tree[i*2+1];
    }
}
void upd(int pos, int d)
{
    pos+=N;
    tree[pos].val=d;
    pos/=2;
    while (pos)
    {
        tree[pos]=tree[pos*2]+tree[pos*2+1];
        pos/=2;
    }
}
node get(int v, int l, int r) {
    if (l > tree[v].r || tree[v].l > r)
        return node();
    if (l <= tree[v].l && tree[v].r <= r)
        return tree[v];
    return get(v * 2, l, r) + get(v * 2 + 1, l, r);
}
signed main() {
```

```
    std::ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n, k;
    cin >> n >> k;
    string s;
    cin >> s;
    for (int i = 0; i < n; ++i) {
        if (i) {
            for (int j = 0; j < 26; ++j)
                dp[i][j] = dp[i - 1][j];
        }
        dp[i][s[i] - 'a']++;
    }
    vector<int> mx_lft(n);
    for (int i = 1; i < n; ++i) {
        int l = 1, r = i + 1;

        while (r - l > 1) {
            int m = (l + r) / 2;
            if (ok(i, m))
                l = m;
            else
                r = m;
        }
        if (ok(i, l + 1))
            l++;
        mx_lft[i] = i - l + 1;
        mx_lft[i] = max(mx_lft[i], i - k + 1);
    }
    build();
    vector<int> ans(n);
    for (int i = 0; i < n; ++i) {
        ans[i] = 1e9;
        int l = mx_lft[i];
        if (l == 0) {
            ans[i] = 1;
            upd(i, 1);
            continue;
        }
        ans[i]=get(1, l-1, i-1).val+1;
        upd(i, ans[i]);
    }
    cout << ans[n - 1];
    return 0;
}
```

Task C ()

```
#include <iostream>
#include <algorithm>
#include <string.h>
#include <cstdlib>
#include <vector>

#define ll long long
using namespace std;
int dp[501][250001];
int v[501], w[501];
char ans[501];
int main() {
    std::ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n, x, y;
    cin >> n >> x >> y;

    for (int i = 0; i < n; ++i)
        cin >> v[i];
    for (int i = 0; i < n; ++i)
        cin >> w[i];
    for (int i = 0; i < n; ++i) {
        if (i == 0) {
            for (int j = 0; j <= x; ++j)
                dp[i][j] = -1;
            //put to the 2 box
            if (w[0] <= y) {
                dp[i][0] = w[0];
            }
            //put to the 1 box
            if (v[i] <= x) {
                dp[i][v[i]] = 0;
            }
            continue;
        }
        for (int j = 0; j <= x; ++j) {
            dp[i][j] = -1;
            if (j >= v[i])//put i to the first box
            {
                if (dp[i - 1][j - v[i]] != -1 && dp[i - 1][j - v[i]] <= y)
                    dp[i][j] = dp[i - 1][j - v[i]];
            }
            // put i to the second box
            if (dp[i - 1][j] != -1 && dp[i - 1][j] + w[i] <= y &&
                (dp[i][j] == -1 || (dp[i][j] != -1 && dp[i - 1][j] + w[i] <= dp[i][j])))
                dp[i][j] = dp[i - 1][j] + w[i];
        }
    }
    for (int j = 0; j <= x; ++j) {
        if (dp[n - 1][j] != -1 && dp[n - 1][j] <= y) {
            int a1 = j;
            for (int i = n - 1; i >= 0; --i) {
                if (i != 0) {
                    if (a1 - v[i] >= 0 && dp[i - 1][a1 - v[i]] == dp[i][a1]) {
                        ans[i] = 'x';
                        a1 -= v[i];
                    } else {
                        ans[i] = 'y';
                    }
                } else {
                    if (a1 == v[i]) {
                        ans[0] = 'x';
                        a1 -= v[i];
                    } else {
                        ans[0] = 'y';
                    }
                }
            }

            for (int i = 0; i < n; ++i) {
                cout << ans[i];
```

```
            }
            return 0;
        }
    }
    cout << -1;
    return 0;
}
```

Task D ()

```cpp
#pragma GCC optimize("Ofast")
#include <iostream>
#include <algorithm>
#include <string.h>
#include <cstdlib>
#include <vector>

#define ll long long
using namespace std;

int main() {
    std::ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n;
    cin >> n;
    string s;
    cin >> s;
    int cnt = 0;
    int a, b, c, d;
    a = b = c = d = 0;
    for (auto u:s) {
        if (u == ')')
            u = '(';
        if (u == ']')
            u = '[';
        if (cnt % 2 == 0) {
            if (u == '[')
                a++;
            else
                c++;
        } else {
            if (u == '[')
                b++;
            else
                d++;

        }

        cnt++;
    }
    int ans=1e9;
    for(int i=0;i<=n;++i)
    {
        int cur=0;

        cur+=abs(i-a);
        cur+=abs(i-b);
        ans=min(ans, cur);

    }
    cout<<ans;
    return 0;
}
```

Task E ()

```cpp
#include <iostream>
#include <algorithm>
#include <string.h>

#include <vector>

#define ll long long
using namespace std;

signed main() {
    std::ios_base::sync_with_stdio(0);
    cin.tie(0);
    string s;
    cin >> s;
    if (s == "add") {
        int q;
        cin >> q;
        while (q--) {
            int n, k;
            cin >> n >> k;
            vector<int> a(k);
            for (int i = 0; i < k; ++i) {
                cin >> a[i];
            }
            if (n == 1000000) {
                cout << 179179 << endl;
            } else if (n == 10) {
                cout << 4 << endl;
            }
        }
    } else if (s == "clear") {
        int q;
        cin >> q;
        while (q--) {
            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 (auto u:a) {
                    if (u != 179179)
                        cout << u << ' ';
                }
                cout << endl;
            } else if (n == 10) {
                for (auto u:a) {
                    if (u != 4)
                        cout << u << ' ';
                }
                cout << endl;
            }
        }
    }
    return 0;
}
```

Task F ()

```
#include <iostream>
#include <algorithm>
#include <string.h>
#include <map>

#include <vector>

#define ll long long
using namespace std;

signed main() {
    std::ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n;
    cin >> n;
    if (n <= 8) {
        cout << 4 << '\n';
        cout << "0 0\n1 0\n1 1\n0 1\n";
        vector<pair<int, int>> ans;
        for (int i = -1; i <= 1; ++i) {
            for (int j = -1; j <= 1; ++j) {
                if (i == 0 && j == 0)
                    continue;
                ans.push_back({i, j});
            }
        }
        for (int i = 0; i < n; ++i)
            cout << ans[i].first << ' ' << ans[i].second << '\n';

        return 0;
    }
    return 0;
}
```