

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 <iostream>

#define ll long long

using namespace std;

int main()
{
    int k;
    cin >> k;
    int cur = 0, r = 0;
    for (int i = 1; i <= k; ++i)
    {
        cur = i + r;
        r = cur / 10;
        cur %= 10;
    }
    cout << cur;
}
```

Task B ()

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    int n, k;
    string s;
    cin >> n >> k >> s;
    int cnt[26] = { 0 };
    int dif = 0, ans = 1, last_st = 0;
    for (int i = 0; i < n; ++i)
    {
        if (cnt[s[i] - 'a'] == 0 || i - last_st == k)
        {
            if (dif == 3 || i - last_st == k)
            {
                ++ans;
                fill(cnt, cnt + 26, 0);
                last_st = i;
                dif = 0;
            }
            ++cnt[s[i] - 'a'];
            ++dif;
        }
    }
    cout << ans;
}
```

Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>

#define INF 2000000000

using namespace std;

int main()
{
    int n, x, y;
    cin >> n >> x >> y;
    vector<int> v(n), w(n);
    for (int i = 0; i < n; ++i) cin >> v[i];
    for (int i = 0; i < n; ++i) cin >> w[i];

    vector<vector<int>> dp(n + 1, vector<int>(x + 1));
    for (int i = 1; i <= n; ++i)
    {
        for (int j = 1; j <= x; ++j)
        {
            dp[i][j] = max(dp[i - 1][j], v[i - 1] <= j ? dp[i - 1][j - v[i - 1]] + w[i - 1] : 0);
        }
    }

    vector<char> res(n);
    int i = n, j = x;
    while (i > 0)
    {
        if (dp[i][j] == dp[i - 1][j])
        {
            res[i - 1] = 'y';
        } else
        {
            res[i - 1] = 'x';
            j -= v[i - 1];
        }
        --i;
    }

    for (int i = 0; i < n; ++i)
        if (res[i] == 'y') y -= w[i];

    if (y < 0)
    {
        cout << -1;
        return 0;
    }

    for (int i = 0; i < n; ++i) cout << res[i];
}
```

Task D ()

```
#include <iostream>
#include <string>
#include <vector>
#include <stack>

using namespace std;

bool check(char a, char b)
{
    return abs(a - b) <= 2;
}

char get_pair(char a)
{
    if (a == '(') return ')';
    if (a == ')') return '(';
    if (a == '[') return ']';
    return '[';
}

int main()
{
    int n;
    string s;
    cin >> n >> s;
    int ans = 0, bal = 0;
    stack<char> st;
    for (int i = 0; i < 2 * n; ++i)
    {
        if (s[i] == ')') || s[i] == ']')
        {
            if (st.empty())
            {
                st.push(get_pair(s[i]));
                ++bal;
            }
            else
            {
                if (check(st.top(), s[i]))
                {
                    st.pop();
                    --bal;
                } else if (2 * n - i <= bal)
                {
                    st.pop();
                    --bal;
                    ++ans;
                } else
                {
                    ++bal;
                    st.push(get_pair(s[i]));
                }
            }
        } else
        {
            if (2 * n - i <= bal)
            {
                --bal;
                if (!check(st.top(), s[i])) ++ans;
                st.pop();
            } else
            {
                if (!st.empty() && check(st.top(), s[i]))
                {
                    st.pop();
                    --bal;
                }
                else
                {
                    st.push(s[i]);
                    ++bal;
                }
            }
        }
    }
}
```

```
    }  
    }  
    cout << ans;  
}
```

Task E ()

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    string s;
    cin >> s;
    int t;
    cin >> t;
    for (int q = 0; q < t; ++q)
    {
        int n, k;
        cin >> n >> k;
        if (s == "add")
        {
            vector<int> v(k + 2);
            for (int i = 0; i < k; ++i) cin >> v[i];
            v[k] = 0;
            v[k + 1] = n + 1;
            sort(v.begin(), v.end());

            int mx_dist = 0, r = 0;

            for (int i = 1; i < v.size(); ++i)
            {
                if (v[i] - v[i - 1] > mx_dist)
                {
                    mx_dist = v[i] - v[i - 1];
                    r = v[i - 1] + 1 + (i == 1);
                }
            }
            cout << r << '\n';
        } else
        {
            vector<int> v(k + 3);
            for (int i = 0; i <= k; ++i) cin >> v[i];
            v[k + 1] = 0;
            v[k + 2] = n + 1;
            sort(v.begin(), v.end());
            int mx_dist = -10, r = 0;
            for (int i = 2; i < v.size(); ++i)
            {
                if (v[i] - v[i - 2] > mx_dist && v[i - 1] == v[i - 2] + 1 + (i == 2))
                {
                    mx_dist = v[i] - v[i - 2];
                    r = v[i - 1];
                }
            }
            for (int i = 1; i < k + 2; ++i)
                if (v[i] != r) cout << v[i] << ' ';
            cout << '\n';
        }
    }
}
```

Task F ()

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    int n;
    cin >> n;
    cout << "4\4\5\5\4\5\n";
    vector<pair<int, int>> v = { {0, -1}, {1, -1}, {1, 0}, {1, 1}, {0, 1}, {-1, 1}, {-1, 0},
        {-1, -1} };
    for (int i = 0; i < n; ++i) cout << v[min(i, 7)].first << '\u' << v[min(i, 7)].second << '\n';
}
```