

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	55	25	480

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <queue>
#include <assert.h>

using namespace std;

typedef long long ll;
typedef unsigned long long ull;
const ll INF = (ll)1e18;
const double PI = acos(-1.0);

int main()
{
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    ll k;
    cin >> k;

    if (k <= 10000)
    {
        ll cur = 0;
        for (ll i = 1; i < k; i++)
        {
            //cout << (i + cur) % 10;
            cur = (i + cur) / 10;
        }
        cout << (k + cur) % 10;
    }
    else
    {
        k -= 10;
        ll ans = k % 9;
        if (ans > 0)
            ans++;
        cout << ans;
    }

    return 0;
}
```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <queue>
#include <assert.h>

using namespace std;

typedef long long ll;
typedef unsigned long long ull;
const ll INF = (ll)1e18;
const double PI = acos(-1.0);

const int MAX_N = 100005;
int n, k;
string s;
int dp[MAX_N];
int mindp[MAX_N][20];

int get_mindp(int l, int r)
{
    int len = r - l + 1;
    int j = 0;
    while ((l << j) <= len)
        j++;
    j--;

    //cout << "get " << r << " " << l + (1 << j) - 1 << " " << j << endl;
    return min(mindp[r][j], mindp[l + (1 << j) - 1][j]);
}

int main()
{
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    cin >> n >> k >> s;
    vector<int> is(26, 0);
    int cnt = 0;
    int l = 0;
    for (int i = 0; i < n; i++)
    {
        if (!is[s[i] - 'a'])
            cnt++;
        is[s[i] - 'a']++;
        while (cnt > 3 || i - l + 1 > k)
        {
            if (is[s[l] - 'a'] == 1)
                cnt--;
            is[s[l] - 'a']--;
            l++;
        }

        if (l == 0)
        {
            dp[i] = 1;
        }
        else
        {
            dp[i] = get_mindp(l - 1, i - 1) + 1;
        }

        //cout << l << " " << i << " " << dp[i] << endl;
    }
}
```

```

mindp[i][0] = dp[i];
for (int j = 1; j < 20; j++)
{
    if ((1 << j) > i + 1)
        break;
    mindp[i][j] = min(mindp[i][j - 1], mindp[i - (1 << (j - 1))][j - 1]);
}

cout << dp[n - 1];

return 0;
}

```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <queue>
#include <assert.h>

using namespace std;

typedef long long ll;
typedef unsigned long long ull;
const ll INF = (ll)1e18;
const double PI = acos(-1.0);

const int MAX_N = 505;
const int MAX_W = 250005;
int n, x, y;
int v[MAX_N];
int w[MAX_N];
int dp[2][MAX_W];
bool ans[MAX_N][MAX_W];

int main()
{
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

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

    for (int j = 0; j <= y; j++)
    {
        dp[0][j] = -1;
        if (j == 0)
        {
            dp[0][j] = 0;
        }
        if (j == w[0])
        {
            dp[0][j] = v[0];
            ans[0][j] = 1;
        }
        //cout << 0 << " " << j << " " << dp[0][j] << endl;
    }
    for (int i = 1; i < n; i++)
    {
        for (int j = 0; j <= y; j++)
        {
            dp[i % 2][j] = dp[1 - i % 2][j];
            if (j - w[i] >= 0 && dp[1 - i % 2][j - w[i]] != -1 && dp[1 - i % 2][j - w[i]] + v[i] >
                dp[i % 2][j])
            {
                dp[i % 2][j] = dp[1 - i % 2][j - w[i]] + v[i];
                ans[i][j] = 1;
            }
        }
    }
}
```

```

        //cout << i << " " << j << " " << dp[i % 2][j] << endl;
    }
}

for (int j = 0; j <= y; j++)
{
    if (dp[(n - 1) % 2][j] != -1 && full - dp[(n - 1) % 2][j] <= x)
    {
        string s = "";
        int cur = j;
        for (int i = n - 1; i >= 0; i--)
        {
            if (ans[i][cur])
            {
                s += 'y';
                cur -= w[i];
            }
            else
            {
                s += 'x';
            }
        }
        reverse(s.begin(), s.end());
        cout << s;

        return 0;
    }
}
cout << -1;

return 0;
}

```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <queue>
#include <assert.h>

using namespace std;

typedef long long ll;
typedef unsigned long long ull;
const ll INF = (ll)1e18;
const double PI = acos(-1.0);

int main()
{
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    int n;
    string s;
    cin >> n >> s;
    n *= 2;
    int cur = 0;
    bool last = 0;
    for (int i = 0; i < n; i++)
    {
        if (s[i] == '(' || s[i] == ')')
        {
            if (cur == 0 || last == 1)
            {
                cur++;
                last = 0;
            }
            else
            {
                cur--;
                last = 1;
            }
        }
        else
        {
            if (cur == 0 || last == 0)
            {
                cur++;
                last = 1;
            }
            else
            {
                cur--;
                last = 0;
            }
        }
    }
    cout << cur / 2;

    return 0;
}
```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <queue>
#include <assert.h>

using namespace std;

typedef long long ll;
typedef unsigned long long ull;
const ll INF = (ll)1e18;
const double PI = acos(-1.0);

const int MAGIC = 120103;
map<pair<int, pair<int, int>>, int> mp;
set<pair<pair<int, int>, pair<int, int>>> st;

bool check(int a, int b, int c, int d)
{
    if (a == d || b == d || c == d)
        return 0;

    pair<int, pair<int, int>> pr = { a, { b, c } };
    int ans = d;
    if (c > d)
        swap(c, d);
    if (b > c)
        swap(b, c);
    if (a > b)
        swap(a, b);

    if (st.find({ { a, b }, { c, d } }) == st.end())
    {
        st.insert({ { a, b }, { c, d } });
        mp[pr] = ans;
        return 1;
    }
    else
    {
        return 0;
    }
}

int main()
{
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    srand(1201);

    for (int a = 1; a <= 10; a++)
    {
        for (int b = a + 1; b <= 10; b++)
        {
            for (int c = b + 1; c <= 10; c++)
            {
                for (int d = 1; d <= 10; d++)
                {
                    if (check(a, b, c, d))
                    {
                        break;
                    }
                    else if (d == 10)
                    {

```

```

        cout << "BAD_" << a << "_" << b << "_" << c << endl;
    }
}
}

vector<int> vr(100000);
for (int i = 0; i < 100000; i++)
{
    vr[i] = i + 1;
}
random_shuffle(vr.begin(), vr.end());

string s;
int t, n, k;
cin >> s >> t;
while (t--)
{
    cin >> n >> k;
    if (s == "add")
    {
        vector<int> v(k);
        for (int i = 0; i < k; i++)
        {
            cin >> v[i];
        }

        if (n == 1000000)
        {
            cout << MAGIC << endl;
        }
        else if (n == 10)
        {
            sort(v.begin(), v.end());
            cout << mp[{v[0], {v[1], v[2]}}] << endl;
        }
        else if (n == 100000)
        {
            vector<bool> used(n + 1, 0);
            for (int i = 0; i < k; i++)
            {
                used[v[i]] = 1;
            }
            for (int i = 0; i < n; i++)
            {
                if (vr[i] + 5 > n)
                    continue;
                if (!used[vr[i]] && !used[vr[i] + 1] && !used[vr[i] + 2] && !used[vr[i] + 3]
                    && !used[vr[i] + 4] && !used[vr[i] + 5])
                {
                    cout << vr[i] << endl;
                    break;
                }
                else if (i == n - 1)
                {
                    cout << 321 << endl;
                }
            }
        }
    }
}
else
{
    vector<int> v(k + 1);
    for (int i = 0; i < k + 1; i++)
    {
        cin >> v[i];
    }

    if (n == 1000000)
    {
        for (int i = 0; i < k + 1; i++)
        {
            if (v[i] != MAGIC)

```



```

        {
            cout << v[i] << " ";
        }
    }
}
else if (n == 10)
{
    sort(v.begin(), v.end());
    do
    {
        if (mp[{v[0], {v[1], v[2]}}] == v[3])
        {
            break;
        }
    } while (next_permutation(v.begin(), v.end()));
    for (int i = 0; i < k; i++)
    {
        cout << v[i] << " ";
    }
}
else if (n == 100000)
{
    vector<bool> used(n + 1, 0);
    for (int i = 0; i < k + 1; i++)
    {
        used[v[i]] = 1;
    }
    int aaa = v[rand() % k];
    for (int i = 0; i < n; i++)
    {
        if (vr[i] + 5 > n)
            continue;
        if (used[vr[i]] && !used[vr[i] + 1] && !used[vr[i] + 2] && !used[vr[i] + 3] &&
            !used[vr[i] + 4] && !used[vr[i] + 5])
        {
            aaa = vr[i];
        }
    }

    for (int i = 0; i < k + 1; i++)
    {
        if (v[i] != aaa)
        {
            cout << v[i] << " ";
        }
    }
}

cout << endl;
}
}

return 0;
}

```

Task F ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <map>
#include <set>
#include <queue>
#include <assert.h>

using namespace std;

typedef long long ll;
typedef unsigned long long ull;
const ll INF = (ll)1e18;
const double PI = acos(-1.0);

int main()
{
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    int n;
    cin >> n;

    cout << 4 << endl;
    cout << "0_0" << endl;
    cout << "0_1" << endl;
    cout << "1_1" << endl;
    cout << "1_0" << endl;

    for (int x = -1; x <= 1; x++)
    {
        for (int y = -1; y <= 1; y++)
        {
            if (x == 0 && y == 0)
                continue;
            if (n <= 0)
                break;

            cout << x << "_" << y << endl;
            n--;
        }
    }

    return 0;
}
```