

# 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 <iomanip>

#define endl '\n'

using namespace std;

long long k;

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    ios_base::sync_with_stdio(0);
    cin.tie(0);

    cin >> k;

    if (k < 10)
    {
        cout << k << endl;
    }
    else
    {
        k--;
        if (k % 9 == 0)
        {
            cout << 0 << endl;
        }
        else
        {
            cout << k % 9 + 1 << endl;
        }
    }
}
```

## Task B ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <iomanip>
#include <string>

#define endl '\n'

using namespace std;

const int Base = 1e5 + 15;

int n, k;
string s;
int ans[Base];
int letterCnt[40];
int difCnt = 0;

void add(int let)
{
    if (letterCnt[let] == 0)
    {
        difCnt++;
    }
    letterCnt[let]++;
}

void del(int let)
{
    if (letterCnt[let] == 1)
    {
        difCnt--;
    }
    letterCnt[let]--;
}

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    ios_base::sync_with_stdio(0);
    cin.tie(0);

    cin >> n >> k;

    cin >> s;

    int l = 0, r = 0;
    ans[0] = 0;
    for (int i = 1; i <= n; i++)
    {
        add(s[r] - 'a');
        r++;
        while (difCnt > 3 || r - l > k)
        {
            del(s[l] - 'a');
            l++;
        }
        ans[i] = ans[l] + 1;
    }

    cout << ans[n] << endl;

    return 0;
}
```

## Task C ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <iomanip>
#include <string>
#include <random>
#include <bitset>

#define endl '\n'

using namespace std;

const int Base = 250005;
mt19937 rnd(777);

int n, x, y;
int w[505];
int c[505];
int sumC = 0;
bitset<505> p[Base];
int dp[Base];

string solution()
{
    for (int i = 0; i <= x; i++)
    {
        dp[i] = 0;
    }

    for (int i = 0; i < n; i++)
    {
        for (int j = x; j >= 0; j--)
        {
            if (j + w[i] <= x)
            {
                if (dp[j + w[i]] < dp[j] + c[i])
                {
                    dp[j + w[i]] = dp[j] + c[i];
                    p[j + w[i]] = p[j];
                    p[j + w[i]].set(i);
                }
            }
        }
    }

    int maxId = 0;
    for (int i = 0; i <= x; i++)
    {
        if (dp[maxId] < dp[i])
        {
            maxId = i;
        }
    }

    if (sumC - dp[maxId] <= y)
    {
        string ans = "";
        for (int i = 0; i < n; i++)
        {
            if (p[maxId][i])
            {
                ans += 'x';
            }
            else
            {
                ans += 'y';
            }
        }
        return ans;
    }
    else
    {

```

```

        return "-1";
    }
}

bool check(string ans)
{
    if (ans == "-1")
    {
        return true;
    }
    else
    {
        int s1 = 0, s2 = 0;
        for (int i = 0; i < n; i++)
        {
            if (ans[i] == 'x')
            {
                s1 += w[i];
            }
            else
            {
                s2 += c[i];
            }
        }

        return (s1 <= x && s2 <= y);
    }
}

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    ios_base::sync_with_stdio(0);
    cin.tie(0);

    cin >> n >> x >> y;

    for (int i = 0; i < n; i++)
    {
        cin >> w[i];
    }
    for (int i = 0; i < n; i++)
    {
        cin >> c[i];
        sumC += c[i];
    }

    cout << solution() << endl;

    while (0)
    {
        n = rnd() % 3 + 1;
        x = rnd() % 15 + 1;
        y = rnd() % 15 + 1;
        for (int i = 0; i < n; i++)
        {
            w[i] = rnd() % 6;
        }
        sumC = 0;
        for (int i = 0; i < n; i++)
        {
            c[i] = rnd() % 6;
            sumC += c[i];
        }
        string ans = solution();

        if (check(ans))
        {
            cout << ans << endl;
        }
        else

```

```

    {
        cout << n << ' ' << x << ' ' << y << endl;
        for (int i = 0; i < n; i++)
        {
            cout << w[i] << ' ';
        }
        cout << endl;
        for (int i = 0; i < n; i++)
        {
            cout << c[i] << ' ';
        }
        cout << endl;
        cout << ans << endl;
        return 0;
    }
}

```

## Task D ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <iomanip>
#include <string>
#include <random>
#include <bitset>

#define endl '\n'

using namespace std;

const int Base = 500;
mt19937 rnd(777);

int n;
string s;
int dp[Base][Base];
int ans[Base];

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    ios_base::sync_with_stdio(0);
    cin.tie(0);

    cin >> n;
    cin >> s;

    for (int i = 0; i < 2 * n; i++)
    {
        if (s[i] == ')')
        {
            s[i] = '(';
        }
        if (s[i] == '[')
        {
            s[i] = ']';
        }
    }

    vector<char> last;
    last.push_back('$');
    for (int i = 0; i < 2 * n; i++)
    {
        if (last.back() == s[i])
        {
            last.pop_back();
        }
        else
        {
            last.push_back(s[i]);
        }
    }

    cout << last.size() / 2 << endl;

    return 0;
}
```

## Task E ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <iomanip>
#include <string>
#include <random>
#include <bitset>
#include <algorithm>

#define endl '\n'

using namespace std;

const int Base = 250005;
mt19937 rnd(777);

const int ourNum = 732562;
int t, n, k;
string type;

void group1()
{
    if (type == "add")
    {
        cout << ourNum << endl;
    }
    else
    {
        for (int i = 0; i < k + 1; i++)
        {
            int num;
            cin >> num;
            if (num != ourNum)
            {
                cout << num << ' ';
            }
        }
        cout << endl;
    }
}

int getAdd(string s)
{
    if (s == "012") {
        return 3;
    }
    if (s == "013") {
        return 4;
    }
    if (s == "014") {
        return 2;
    }
    if (s == "015") {
        return 2;
    }
    if (s == "016") {
        return 2;
    }
    if (s == "017") {
        return 2;
    }
    if (s == "018") {
        return 2;
    }
    if (s == "019") {
        return 2;
    }
    if (s == "023") {
        return 4;
    }
    if (s == "024") {
        return 5;
    }
}
```

```

}
if (s == "025") {
    return 3;
}
if (s == "026") {
    return 3;
}
if (s == "027") {
    return 3;
}
if (s == "028") {
    return 3;
}
if (s == "029") {
    return 3;
}
if (s == "034") {
    return 5;
}
if (s == "035") {
    return 1;
}
if (s == "036") {
    return 1;
}
if (s == "037") {
    return 1;
}
if (s == "038") {
    return 1;
}
if (s == "039") {
    return 1;
}
if (s == "045") {
    return 1;
}
if (s == "046") {
    return 1;
}
if (s == "047") {
    return 1;
}
if (s == "048") {
    return 1;
}
if (s == "049") {
    return 1;
}
if (s == "056") {
    return 1;
}
if (s == "057") {
    return 1;
}
if (s == "058") {
    return 1;
}
if (s == "059") {
    return 1;
}
if (s == "067") {
    return 1;
}
if (s == "068") {
    return 1;
}
if (s == "069") {
    return 1;
}
if (s == "078") {
    return 1;
}
if (s == "079") {

```



```

        return 1;
    }
    if (s == "089") {
        return 1;
    }
    if (s == "123") {
        return 4;
    }
    if (s == "124") {
        return 5;
    }
    if (s == "125") {
        return 3;
    }
    if (s == "126") {
        return 3;
    }
    if (s == "127") {
        return 3;
    }
    if (s == "128") {
        return 3;
    }
    if (s == "129") {
        return 3;
    }
    if (s == "134") {
        return 5;
    }
    if (s == "135") {
        return 6;
    }
    if (s == "136") {
        return 4;
    }
    if (s == "137") {
        return 4;
    }
    if (s == "138") {
        return 4;
    }
    if (s == "139") {
        return 4;
    }
    if (s == "145") {
        return 6;
    }
    if (s == "146") {
        return 2;
    }
    if (s == "147") {
        return 2;
    }
    if (s == "148") {
        return 2;
    }
    if (s == "149") {
        return 2;
    }
    if (s == "156") {
        return 2;
    }
    if (s == "157") {
        return 2;
    }
    if (s == "158") {
        return 2;
    }
    if (s == "159") {
        return 2;
    }
    if (s == "167") {
        return 2;
    }
}

```

```

if (s == "168") {
    return 2;
}
if (s == "169") {
    return 2;
}
if (s == "178") {
    return 2;
}
if (s == "179") {
    return 2;
}
if (s == "189") {
    return 2;
}
if (s == "234") {
    return 5;
}
if (s == "235") {
    return 6;
}
if (s == "236") {
    return 4;
}
if (s == "237") {
    return 4;
}
if (s == "238") {
    return 4;
}
if (s == "239") {
    return 4;
}
if (s == "245") {
    return 6;
}
if (s == "246") {
    return 0;
}
if (s == "247") {
    return 0;
}
if (s == "248") {
    return 0;
}
if (s == "249") {
    return 0;
}
if (s == "256") {
    return 0;
}
if (s == "257") {
    return 0;
}
if (s == "258") {
    return 0;
}
if (s == "259") {
    return 0;
}
if (s == "267") {
    return 0;
}
if (s == "268") {
    return 0;
}
if (s == "269") {
    return 0;
}
if (s == "278") {
    return 0;
}
if (s == "279") {
    return 0;
}

```

```

}
if (s == "289") {
    return 0;
}
if (s == "345") {
    return 6;
}
if (s == "346") {
    return 0;
}
if (s == "347") {
    return 0;
}
if (s == "348") {
    return 0;
}
if (s == "349") {
    return 0;
}
if (s == "356") {
    return 0;
}
if (s == "357") {
    return 0;
}
if (s == "358") {
    return 0;
}
if (s == "359") {
    return 0;
}
if (s == "367") {
    return 0;
}
if (s == "368") {
    return 0;
}
if (s == "369") {
    return 0;
}
if (s == "378") {
    return 0;
}
if (s == "379") {
    return 0;
}
if (s == "389") {
    return 0;
}
if (s == "456") {
    return 0;
}
if (s == "457") {
    return 0;
}
if (s == "458") {
    return 0;
}
if (s == "459") {
    return 0;
}
if (s == "467") {
    return 0;
}
if (s == "468") {
    return 0;
}
if (s == "469") {
    return 0;
}
if (s == "478") {
    return 0;
}
if (s == "479") {

```

```

        return 0;
    }
    if (s == "489") {
        return 0;
    }
    if (s == "567") {
        return 0;
    }
    if (s == "568") {
        return 0;
    }
    if (s == "569") {
        return 0;
    }
    if (s == "578") {
        return 0;
    }
    if (s == "579") {
        return 0;
    }
    if (s == "589") {
        return 0;
    }
    if (s == "678") {
        return 0;
    }
    if (s == "679") {
        return 0;
    }
    if (s == "689") {
        return 0;
    }
    if (s == "789") {
        return 0;
    }
}

int getDel(string s)
{
    if (s == "0123") {
        return 3;
    }
    if (s == "0134") {
        return 4;
    }
    if (s == "0124") {
        return 2;
    }
    if (s == "0125") {
        return 2;
    }
    if (s == "0126") {
        return 2;
    }
    if (s == "0127") {
        return 2;
    }
    if (s == "0128") {
        return 2;
    }
    if (s == "0129") {
        return 2;
    }
    if (s == "0234") {
        return 4;
    }
    if (s == "0245") {
        return 5;
    }
    if (s == "0235") {
        return 3;
    }
    if (s == "0236") {

```

```

        return 3;
    }
    if (s == "0237") {
        return 3;
    }
    if (s == "0238") {
        return 3;
    }
    if (s == "0239") {
        return 3;
    }
    if (s == "0345") {
        return 5;
    }
    if (s == "0135") {
        return 1;
    }
    if (s == "0136") {
        return 1;
    }
    if (s == "0137") {
        return 1;
    }
    if (s == "0138") {
        return 1;
    }
    if (s == "0139") {
        return 1;
    }
    if (s == "0145") {
        return 1;
    }
    if (s == "0146") {
        return 1;
    }
    if (s == "0147") {
        return 1;
    }
    if (s == "0148") {
        return 1;
    }
    if (s == "0149") {
        return 1;
    }
    if (s == "0156") {
        return 1;
    }
    if (s == "0157") {
        return 1;
    }
    if (s == "0158") {
        return 1;
    }
    if (s == "0159") {
        return 1;
    }
    if (s == "0167") {
        return 1;
    }
    if (s == "0168") {
        return 1;
    }
    if (s == "0169") {
        return 1;
    }
    if (s == "0178") {
        return 1;
    }
    if (s == "0179") {
        return 1;
    }
    if (s == "0189") {
        return 1;
    }
}

```

```

if (s == "1234") {
    return 4;
}
if (s == "1245") {
    return 5;
}
if (s == "1235") {
    return 3;
}
if (s == "1236") {
    return 3;
}
if (s == "1237") {
    return 3;
}
if (s == "1238") {
    return 3;
}
if (s == "1239") {
    return 3;
}
if (s == "1345") {
    return 5;
}
if (s == "1356") {
    return 6;
}
if (s == "1346") {
    return 4;
}
if (s == "1347") {
    return 4;
}
if (s == "1348") {
    return 4;
}
if (s == "1349") {
    return 4;
}
if (s == "1456") {
    return 6;
}
if (s == "1246") {
    return 2;
}
if (s == "1247") {
    return 2;
}
if (s == "1248") {
    return 2;
}
if (s == "1249") {
    return 2;
}
if (s == "1256") {
    return 2;
}
if (s == "1257") {
    return 2;
}
if (s == "1258") {
    return 2;
}
if (s == "1259") {
    return 2;
}
if (s == "1267") {
    return 2;
}
if (s == "1268") {
    return 2;
}
if (s == "1269") {
    return 2;
}

```

```

}
if (s == "1278") {
    return 2;
}
if (s == "1279") {
    return 2;
}
if (s == "1289") {
    return 2;
}
if (s == "2345") {
    return 5;
}
if (s == "2356") {
    return 6;
}
if (s == "2346") {
    return 4;
}
if (s == "2347") {
    return 4;
}
if (s == "2348") {
    return 4;
}
if (s == "2349") {
    return 4;
}
if (s == "2456") {
    return 6;
}
if (s == "0246") {
    return 0;
}
if (s == "0247") {
    return 0;
}
if (s == "0248") {
    return 0;
}
if (s == "0249") {
    return 0;
}
if (s == "0256") {
    return 0;
}
if (s == "0257") {
    return 0;
}
if (s == "0258") {
    return 0;
}
if (s == "0259") {
    return 0;
}
if (s == "0267") {
    return 0;
}
if (s == "0268") {
    return 0;
}
if (s == "0269") {
    return 0;
}
if (s == "0278") {
    return 0;
}
if (s == "0279") {
    return 0;
}
if (s == "0289") {
    return 0;
}
if (s == "3456") {

```

```

        return 6;
    }
    if (s == "0346") {
        return 0;
    }
    if (s == "0347") {
        return 0;
    }
    if (s == "0348") {
        return 0;
    }
    if (s == "0349") {
        return 0;
    }
    if (s == "0356") {
        return 0;
    }
    if (s == "0357") {
        return 0;
    }
    if (s == "0358") {
        return 0;
    }
    if (s == "0359") {
        return 0;
    }
    if (s == "0367") {
        return 0;
    }
    if (s == "0368") {
        return 0;
    }
    if (s == "0369") {
        return 0;
    }
    if (s == "0378") {
        return 0;
    }
    if (s == "0379") {
        return 0;
    }
    if (s == "0389") {
        return 0;
    }
    if (s == "0456") {
        return 0;
    }
    if (s == "0457") {
        return 0;
    }
    if (s == "0458") {
        return 0;
    }
    if (s == "0459") {
        return 0;
    }
    if (s == "0467") {
        return 0;
    }
    if (s == "0468") {
        return 0;
    }
    if (s == "0469") {
        return 0;
    }
    if (s == "0478") {
        return 0;
    }
    if (s == "0479") {
        return 0;
    }
    if (s == "0489") {
        return 0;
    }
}

```



```

        if (s == "0567") {
            return 0;
        }
        if (s == "0568") {
            return 0;
        }
        if (s == "0569") {
            return 0;
        }
        if (s == "0578") {
            return 0;
        }
        if (s == "0579") {
            return 0;
        }
        if (s == "0589") {
            return 0;
        }
        if (s == "0678") {
            return 0;
        }
        if (s == "0679") {
            return 0;
        }
        if (s == "0689") {
            return 0;
        }
        if (s == "0789") {
            return 0;
        }
    }

}

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    ios_base::sync_with_stdio(0);
    cin.tie(0);

    cin >> type;
    cin >> t >> n >> k;

    if (n == 1e6)
    {
        group1();
    }
    else
    {
        while (t--)
        {
            int r = 3;
            if (type != "add")
            {
                r = 4;
            }
            string s = "";
            for (int i = 0; i < r; i++)
            {
                int key;
                cin >> key;
                key--;
                s.push_back(key + '0');
            }
            sort(s.begin(), s.end());
            if (type == "add")
            {
                cout << getAdd(s) + 1 << endl;
            }
            else
            {
                int our = getDel(s);
                for (int i = 0; i < r; i++)

```

```

        {
            if (s[i] - '0' != our)
            {
                cout << s[i] - '0' + 1 << '␣';
            }
        }
        cout << endl;
    }
    if (t != 0)
    {
        cin >> n >> k;
    }
}

return 0;
}

```

## Task F ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <iomanip>
#include <string>
#include <random>
#include <bitset>

#define endl '\n'

using namespace std;

const int Base = 250005;
mt19937 rnd(777);

int n;
const int m = 4;
const pair<int, int> points[m] = { {0, 0}, {1, 1}, {0, 2}, {-1, 1} };
pair<int, int> vec[8] = { {1, 1}, {2, 0}, {-1, -1}, {0, 2}, {0, -2}, {-2, 0}, {1, -1}, {-1, 1} };

void printPark()
{
    cout << m << endl;
    for (int i = 0; i < m; i++)
    {
        cout << points[i].first << '␣' << points[i].second << endl;
    }
    cout << endl;
}

void printVec(int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << vec[i].first << '␣' << vec[i].second << endl;
    }
    cout << endl;
}

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    ios_base::sync_with_stdio(0);
    cin.tie(0);

    cin >> n;

    printPark();
    printVec(n);

    return 0;
}
```