Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	100	25	525

Task A ()

```
# s = 1
# per = 0
# for i in range(40):
#     print(i + 1, (s + per) % 10)
#     per = (s + per) // 10
#     s += 1
n = int(input())
if n <= 9:
    print(n)
else:
    print([0, 2, 3, 4, 5, 6, 7, 8, 9][(n - 10) % 9])
```

Task B ()

```
#include <iostream>
#include <vector>
#include <bitset>
#include <set>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <unordered_set>
//#include <ext/pb_ds/assoc_container.hpp>
#include <queue>
#include <tuple>
#include <deque>
#include <cassert>
#include <fstream>
using namespace std;
//using namespace __gnu_pbds;
//#pragma GCC optimize("Ofast")

#define int long long
#define sz(x) (int)((x).size())
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define x first
//#define mid ((vl + vr) / 2)
#define lson (2 * v), vl, mid
#define rson (2 * v + 1), mid, vr
#define basic 1, 0, n
#define y second

typedef pair <int, int> pii;
typedef long double ld;
typedef long long ll;
//typedef tree<int, null_type, less <>, rb_tree_tag, tree_order_statistics_node_update> ord_set;

const int N = 3e3 + 4;
const int M = 1e6 + 5;
const int INF = 1e9;
const int MOD = 1e9 + 7;



int32_t main()
{
    ios_base :: sync_with_stdio(0);
    cin.tie(0);
    int n, k;
    string s;
    cin >> n >> k >> s;
    int ans = 0;
    for (int i = 0; i < s.size();)
    {
        vector <char> can;
        int le = 0;
        for (; i < s.size(); ++i)
        {
            bool found = (find(all(can), s[i]) == can.end());
            if ((can.size() == 3 && found) || le == k)
                break;
            if (found)
                can.push_back(s[i]);
            ++le;
        }
        ++ans;
    }
    cout << ans << endl;
}
```

Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
//#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
//using namespace __gnu_pbds;
//#pragma GCC optimize("Ofast")

//#define int long long
#define sz(x) (int)((x).size())
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define x first
//#define mid ((vl + vr) / 2)
#define lson (2 * v), vl, mid
#define rson (2 * v + 1), mid, vr
#define basic 1, 0, n
#define y second

typedef pair <int, int> pii;
typedef long double ld;
typedef long long ll;
//typedef tree<int, null_type, less <>, rb_tree_tag, tree_order_statistics_node_update> ord_set;

const int N = 2.5e5 + 4;
const int M = 505;
const int INF = 1e9;
const int MOD = 1e9 + 7;

int dp[2][N];
int parent[M][N];
int V[M];
int W[M];

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n, vmax, wmax;
    cin >> n >> vmax >> wmax;
    for (int i = 0; i < n; ++i)
        cin >> V[i];
    for (int i = 0; i < n; ++i)
        cin >> W[i];
    dp[0][0] = 0;
    fill(dp[0] + 1, dp[0] + N, INF);
    for (int i = 0; i < n; ++i)
    {
        int cur = 1 - (i & 1);
        int prev = 1 - cur;
        for (int curv = 0; curv < V[i]; ++curv)
        {
            dp[cur][curv] = dp[prev][curv] + W[i];
            parent[i][curv] = curv;
        }
        for (int curv = V[i]; curv < N; ++curv)
        {
            if (dp[prev][curv] + W[i] > dp[prev][curv - V[i]])
            {
                dp[cur][curv] = dp[prev][curv - V[i]];
                parent[i][curv] = curv - V[i];
            }
            else
            {
                dp[cur][curv] = dp[prev][curv] + W[i];
                parent[i][curv] = curv;
            }
        }
    }
    for (int i = 0; i <= vmax; ++i)
    {
        if (dp[n & 1][i] <= wmax)
        {
            vector <char> ans;
            int curv = i;
            for (int j = n - 1; j > -1; --j)
            {
                if (parent[j][curv] == curv)
                    ans.push_back('y');
                else
                {
                    ans.push_back('x');
                    curv = parent[j][curv];
                }
            }
            reverse(all(ans));
            for (auto el : ans)
                cout << el;
            cout << endl;
            return 0;
        }
    }
    cout << -1 << endl;
}
```

Task D ()

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
//#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
//using namespace __gnu_pbds;
//#pragma GCC optimize("Ofast")

//#define int long long
#define sz(x) (int)((x).size())
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define x first
//#define mid ((vl + vr) / 2)
#define lson (2 * v), vl, mid
#define rson (2 * v + 1), mid, vr
#define basic 1, 0, n
#define y second

typedef pair <int, int> pii;
typedef long double ld;
typedef long long ll;
//typedef tree<int, null_type, less <>, rb_tree_tag, tree_order_statistics_node_update> ord_set;

const int N = 2.5e5 + 4;
const int M = 505;
const int INF = 1e9;
const int MOD = 1e9 + 7;

int dp[2][N];
int parent[M][N];
int V[M];
int W[M];

char conv(char x)
{
    if (x == '(' || x == '[')
        return x;
    if (x == ')')
        return '(';
    return '[';
}

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    int n;
    string s;
    cin >> n >> s;
    vector <char> stack;
    for (char el : s)
    {
        el = conv(el);
        if (stack.size() && stack.back() == el)
            stack.pop_back();
        else
            stack.push_back(el);
    }
    cout << stack.size() / 2;
}
```

Task E ()

```cpp
#include <iostream>
#include <vector>
#include <cassert>
#include <algorithm>
#include <random>
//#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
//using namespace __gnu_pbds;
//#pragma GCC optimize("Ofast")

//#define int long long
#define sz(x) (int)((x).size())
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define x first
//#define mid ((vl + vr) / 2)
#define lson (2 * v), vl, mid
#define rson (2 * v + 1), mid, vr
#define basic 1, 0, n
#define y second

typedef pair <int, int> pii;
typedef long double ld;
typedef long long ll;
//typedef tree<int, null_type, less <>, rb_tree_tag, tree_order_statistics_node_update> ord_set;

const int N = 1e6 + 1;
const int M = 505;
const int INF = 1e9;
const int RANDSEED = 1;
const int MOD = 1e9 + 7;
const int MASKSIZE = 24;

//mt19937 rng(RANDSEED);

int n, t, k;

vector <int> good;
int k_to_k1[1 << MASKSIZE];
int k1_to_k[1 << MASKSIZE];
int ff[N];
vector <int> rrr = {811540970, 1308204276, 429922489, 1346450278, 574320640, 1421819965,
    1055921695, 1437021299, 853119320, 1815158510, 1795381155, 179193606, 2038242277, 3372989,
    1035047906, 776715499, 975885505, 266458238, 1492121890, 42478816, 1496614798, 2100944841,
    1202426832, 430430017, 1585731352, 938514628, 1166954103, 1033975982, 1324960070, 1155593543,
    65578395, 1787719692, 2119665721, 169902852, 1331328690, 2040833967, 131036693, 1252328738,
    619536864, 43847827, 2030003657, 1035128161, 950700329, 1877899942, 1936633655, 339265963,
    322472567, 383086, 47616572, 984681585, };
    int sz;
int ptr = 0;
int rng()
{
    return rrr[ptr++];
}

void gen_numbers()
{
    if (n == 10)
        for (int i = 1; i <= 10; ++i)
        {
            good.push_back(i);
            ff[i] = i - 1;
        }
    else
    {
        while (good.size() < MASKSIZE)
        {
            int tmp = rng() % n + 1;
            if (find(all(good), tmp) == good.end())
                good.push_back(tmp);
        }
        sort(all(good));
        for (int i = 0; i < good.size(); ++i)
            ff[good[i]] = i;
    }
    sz = good.size();
}

void gen_relation()
{

    vector <int> masks[sz + 1];
    for (int i = 0; i < (1 << sz); ++i)
        masks[__builtin_popcount(i)].push_back(i);
    for (int s = 0; s < sz; ++s)
    {
        for (int el : masks[s])
        {
            bool f = false;
            for (int i = 0; i < sz; ++i)
            {
                if ((el ^ (1 << i)) > el)
                {
                    if (k1_to_k[el ^ (1 << i)] == -1)
                    {
                        k_to_k1[el] = (el ^ (1 << i));
                        k1_to_k[el ^ (1 << i)] = el;
                        f = true;
                        break;
                    }
                }
            }
//            if (!f)
//                cout << s << " BAD\n";
        }
    }
}

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    string s;
    cin >> s >> t;
    cin >> n >> k;
    if (s == "clear")
        ++k;
    vector <int> a;
    for (int i = 0; i < k; ++i)
    {
        int tmp;
        cin >> tmp;
        a.push_back(tmp);
    }
    fill(k_to_k1, k_to_k1 + (1 << MASKSIZE), -1);
    fill(k1_to_k, k1_to_k + (1 << MASKSIZE), -1);
    fill(ff, ff + N, -1);
    gen_numbers();
    gen_relation();
    for (int _ = 0; _ < t; ++_)
    {
        int msk = 0;
        for (int el : a)
            if (ff[el] != -1)
                msk += (1 << ff[el]);
        if (s[0] == 'a')
        {
            if (k_to_k1[msk] == -1)
                assert(false);
            else
                cout << good[__builtin_ctz((k_to_k1[msk]) ^ msk)] << endl;
        }
        else
        {
            if (k1_to_k[msk] == -1)
                assert(false);
            else
            {
                int tmp = good[__builtin_ctz(k1_to_k[msk] ^ msk)];
                for (int el : a)
                    if (el != tmp)
                        cout << el << '␣';
                cout << endl;
            }
        }
        if (_ < t - 1)
        {
            a.clear();
            cin >> n >> k;
            if (s[0] == 'c')
                ++k;
            for (int i = 0; i < k; ++i)
            {
                int tmp;
                cin >> tmp;
                a.push_back(tmp);
            }
        }
    }

}
```

Task F ()

```
s = [(0, 0), (0, 1), (1, 1), (1, 0)]
di = [(i, j) for i in range(-1, 2) for j in range(-1, 2)]
n = int(input())
print(4)
for el in s:
    print(*el)
for el in di:
    if el != (0, 0):
        print(*el)
        n -= 1
        if not n:
            break
```