Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	80	60	52	65	457

Task A ()

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int d[6];
    vector<int> perm;
    for (int i = 0; i < 6; i++) cin >> d[i], perm.push_back(i + 1);
    do {
        vector<int> current;
        bool good = true;
        for (int i = 0; i < 6; i++) {
            int x = perm[i];
            current.push_back(x);
            sort(current.begin(), current.end());
            if (find(current.begin(), current.end(), x) - current.begin() != d[i] - 1) {
                good = false;
                break;
            }
        }
        if (good) {
            for (auto x : perm) {
                cout << x << ' ';
            }
            return 0;
        }
    } while (next_permutation(perm.begin(), perm.end()));
}
```

Task B ()

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    string format;
    cin >> format;
    long long n;
    cin >> n;
    long long s = 0;
    for (int i = 0; i < n; i++) {
        long long x;
        cin >> x;
        s += x;
    }
    if (format == "first") {
        cout << s * 100 * n;
    } else {
        cout << s % (100 * n * n) + s / (100 * n * n);
    }
}
```

Task C ()

```
a = tuple(sorted(map(int, input().split())))
b = tuple(sorted(map(int, input().split())))
c = tuple(sorted(map(int, input().split())))


d = [a, b, c]

d.sort(key=lambda x: x[0] * x[1])

s = set()

def O(s):
    return tuple(sorted(s))

def R(s):
    return ((s[1], s[0]))

def SIMPLE(func):
    for x in [[0, 1, 2], [0, 2, 1], [1, 0, 2], [1, 2, 0], [2, 0, 1], [2, 1, 0]]:
        func(d[x[0]], d[x[1]], d[x[2]])

        func(R(d[x[0]]), d[x[1]], d[x[2]])
        func(d[x[0]], R(d[x[1]]), d[x[2]])
        func(d[x[0]], d[x[1]], R(d[x[2]]))


        func(R(d[x[0]]), d[x[1]], R(d[x[2]]))
        func(d[x[0]], R(d[x[1]]), R(d[x[2]]))
        func(R(d[x[0]]), R(d[x[1]]), d[x[2]])

        func(R(d[x[0]]), R(d[x[1]]), R(d[x[2]]))

s.add(d[-1])

def check(a, b):
    a, b, c, d = a[0], a[1], b[0], b[1]
    if a == c and b < d:
        s.add(O((a, b)))
        s.add(O((d - b, a)))
    if b == d and a < c:
        s.add(O((a, b)))
        s.add(O((c - a, b)))
    if a < c and b < d:
        s.add(O((a, b)))

check(d[0], d[2])
check(d[0], d[1])
check(d[1], d[0])
check(d[1], d[2])
check(d[2], d[0])
check(d[2], d[1])

check(R(d[0]), d[2])
check(R(d[0]), d[1])
check(R(d[1]), d[0])
check(R(d[1]), d[2])
check(R(d[2]), d[0])
check(R(d[2]), d[1])

check(R(d[0]), R(d[2]))
check(R(d[0]), R(d[1]))
check(R(d[1]), R(d[0]))
check(R(d[1]), R(d[2]))
check(R(d[2]), R(d[0]))
check(R(d[2]), R(d[1]))

check(d[0], R(d[2]))
check(d[0], R(d[1]))
check(d[1], R(d[0]))
check(d[1], R(d[2]))
check(d[2], R(d[0]))
check(d[2], R(d[1]))
```

```
def check2(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if c <= f and a < f and a + c > f and b + d <= e:
        s.add(O((b, f - a)))

SIMPLE(check2)

def check_two_sum_left(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if a + d == f and b == c and b < e:
        s.add(O((e - b, f)))

SIMPLE(check_two_sum_left)

def check_two_sum_bot(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if b + d == e and a == c and a < f:
        s.add(O((f - a, e)))

SIMPLE(check_two_sum_bot)

def corners(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if a + c == e and b + d == f:
        s.add(O((a, d)))
        s.add(O((b, c)))

SIMPLE(corners)

def corners2(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if b + d == e and a + c > f and a <= f and c <= f:
        s.add(O((b, f - a)))
        s.add(O((d, f - c)))

SIMPLE(corners2)

def next2(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if a == c == f and b + d <= e:
        s.add(O((e - b - d, a)))

SIMPLE(next2)

def next3(a, b, c):
    a, b, c, d, e, f = a[0], a[1], b[0], b[1], c[0], c[1]
    if a < f and d == f and b + c <= e:
        s.add(O((e - b - c, d)))
SIMPLE(next3)

s = list(s)
s = list(sorted(filter(lambda x: x[0] != 0 and x[1] != 0, s)))
for el in s:
    print(*el)
```

Task D ()

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <deque>
using namespace std;

const int MAX_T = 51;
int t[MAX_T];
int dp[MAX_T][MAX_T][MAX_T][8];
pair<int, int> who[MAX_T][MAX_T][MAX_T][8];

int f(int a, int b, int c, int mode) {
    if (dp[a][b][c][mode] != -1) {
        return dp[a][b][c][mode];
    }
    if (a == 0 && b == 0 && c == 0 && mode == 0) {
        dp[a][b][c][mode] = 0;
        return 0;
    }
    int win = 0;
    if (mode & 1) {
        if (f(t[0], b, c, mode ^ 1) == 0) {
            win = 1;
            who[a][b][c][mode] = {1, 0};
            dp[a][b][c][mode] = 1;
            return dp[a][b][c][mode];
        }
    }
    if (mode & 2) {
        if (f(a, t[1], c, mode ^ 2) == 0) {
            win = 1;
            who[a][b][c][mode] = {2, 0};
            dp[a][b][c][mode] = 1;
            return dp[a][b][c][mode];
        }
    }
    if (mode & 4) {
        if (f(a, b, t[2], mode ^ 4) == 0) {
            win = 1;
            who[a][b][c][mode] = {3, 0};
            dp[a][b][c][mode] = 1;
            return dp[a][b][c][mode];
        }
    }
    for (int x = 1; x <= a; x++) {
        if (win == 0 && f(a - x, b, c, mode) == 0) {
            win = 1;
            who[a][b][c][mode] = {1, x};
            break;
        }
    }
    for (int x = 1; x <= b; x++) {
        if (win == 0 && f(a, b - x, c, mode) == 0) {
            win = 1;
            who[a][b][c][mode] = {2, x};
            break;
        }
    }
    for (int x = 1; x <= c; x++) {
        if (win == 0 && f(a, b, c - x, mode) == 0) {
            win = 1;
            who[a][b][c][mode] = {3, x};
            break;
        }
    }
    dp[a][b][c][mode] = win;
    return win;
}

int main() {
```

```
int n;
cin >> n;
int a = 0, b = 0, c = 0;
int mode = 0;
for (int i = 0; i < n; i++) {
    cin >> t[i];
    mode |= (1 << i);
    if (i == 0) a = t[i];
    if (i == 1) b = t[i];
    if (i == 2) c = t[i];
}
fill(&dp[0][0][0][0], &dp[0][0][0][0] + MAX_T * MAX_T * MAX_T * 8, -1);
f(a, b, c, mode);
if (dp[a][b][c][mode] == 0) {
    cout << -1 << ' ' << -1 << endl;
    return 0;
}
while (true) {
    auto [t1, t2] = who[a][b][c][mode];
    cout << t1 << ' ' << t2 << endl;
    if (t1 == 1) {
        if (t2 == 0) {
            mode ^= 1;
            a = t[0];
        } else {
            a -= t2;
        }
    }
    if (t1 == 2) {
        if (t2 == 0) {
            mode ^= 2;
            b = t[1];
        } else {
            b -= t2;
        }
    }
    if (t1 == 3) {
        if (t2 == 0) {
            mode ^= 4;
            c = t[2];
        } else {
            c -= t2;
        }
    }
    cin >> t1 >> t2;
    if (t1 == 1) {
        if (t2 == 0) {
            mode ^= 1;
            a = t[0];
        } else {
            a -= t2;
        }
    }
    if (t1 == 2) {
        if (t2 == 0) {
            mode ^= 2;
            b = t[1];
        } else {
            b -= t2;
        }
    }
    if (t1 == 3) {
        if (t2 == 0) {
            mode ^= 4;
            c = t[2];
        } else {
            c -= t2;
        }
    }
    if (t1 == -1 && t2 == -1) {
        return 0;
    }
}
```

```
}
```

Task E ()

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <deque>
using namespace std;

void transmit(int t) {
    for (int q = 0; q < t; q++) {
        int n;
        cin >> n;
        vector<string> w;
        string s(10, '0');
        for (int i = 0; i <= 10; s[i++] = '1') {
            if (n & (1 << i)) {
                w.push_back(s);
            }
        }
        while (w.size() < 10) {
            w.push_back(w.back());
        }
        for (auto s : w) {
            cout << s << "\n";
        }
    }
}

void receive(int t) {
    for (int q = 0; q < t; q++) {
        vector<string> w(10);
        for (int i = 0; i < w.size(); i++) {
            cin >> w[i];
        }
        int n = 0;
        for (auto s : w) {
            int cnt = 0;
            for (auto c : s) {
                if (c == '1') {
                    cnt++;
                }
            }
            n |= (1 << cnt);
        }
        cout << n << '\n';
    }
}

int main() {
    int t;
    string s;
    cin >> t >> s;
    if (s == "transmit") {
        transmit(t);
    }
    else {
        receive(t);
    }
    return 0;
}
```

Task F ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <deque>
using namespace std;

deque<pair<int, int>> getBlocks(string& a) {
    deque<pair<int, int>> res;
    int pos = a.size() - 1;
    while (pos != -1) {
        if (a[pos] != ')') {
            res.push_front({a[pos] - '0', 1});
            pos -= 1;
        } else {
            int value = 0, digit = 0;
            pos--;
            int pw = 1;
            while (a[pos] != '|') {
                value = value + pw * (a[pos] - '0');
                pw *= 10;
                pos--;
            }
            pw = 1;
            pos--;
            while (a[pos] != '(') {
                digit = digit + (a[pos] - '0') * pw;
                pw *= 10;
                pos--;
            }
            pos--;
            res.push_front({digit, value});
        }
    }
    /*cout << a << endl;
    for (auto [x, y] : res) {
        cout << x << ' ' << y << endl;
    }*/
    return res;
}

string f(string& a, string& b) {
    deque<pair<int, int>> aBlocks = getBlocks(a);
    deque<pair<int, int>> bBlocks = getBlocks(b);
    int i = aBlocks.size() - 1;
    int j = bBlocks.size() - 1;
    deque<pair<int, int>> res;
    int flag = 0;
    //cout << "LETS GO" << endl;
    while (true) {
        if (i == -1 && j == -1) {
            if (flag == 1) {
                res.push_front({1, 1});
            }
            break;
        }
        int lenA = (i == -1 ? 1e9 : aBlocks[i].second);
        int lenB = (j == -1 ? 1e9 : bBlocks[j].second);
        int digitA = (i == -1 ? 0 : aBlocks[i].first);
        int digitB = (j == -1 ? 0 : bBlocks[j].first);
        int len = min(lenA, lenB);
        if (len > 1 && ((flag + digitA + digitB) % 10) == ((flag + digitA + digitB - 10 >= 0 ? +1
            : 0) + digitA + digitB) % 10) {
            res.push_front({(((flag + digitA + digitB) % 10), len});
            flag = (flag + digitA + digitB - 10 >= 0 ? +1 : 0);
            lenA -= len;
            lenB -= len;

        } else {
            res.push_front({(digitA + digitB + flag) % 10, 1});
            flag = (digitA + digitB + flag) / 10;
```

```
            lenA -= 1;
            lenB -= 1;
        }
        if (i != -1) aBlocks[i].second = lenA;
        if (j != -1) bBlocks[j].second = lenB;
        if (lenA == 0) {
            i--;
        }
        if (lenB == 0) {
            j--;
        }
    }
    string resStr = "";
    for (auto [digit, val] : res) {
        if (val > 1) {
            resStr += "(" + to_string(digit) + "|" + to_string(val) + ")";
        } else {
            resStr += to_string(digit);
        }
    }
    return resStr;
}

int main() {
    string a, b;
    cin >> a >> b;
    string res = f(a, b);
    cout << res;
}
```