

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	52	10	422

Task A ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <map>
#include <unordered_map>
#include <random>
#include <chrono>

using namespace std;
using ll = long long;
using ld = long double;

int main() {
    vector<int> a(6);
    for (int& i : a) {
        cin >> i;
    }
    vector<int> ans;
    for (int i = 0; i < 6; ++i) {
        for (int& j : ans) {
            if (j >= a[i]) {
                j++;
            }
        }
        ans.push_back(a[i]);
    }
    for (int i : ans) cout << i << " ";
    return 0;
}
```

Task B ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>
#include <map>
#include <unordered_map>
#include <random>
#include <chrono>

using namespace std;
using ll = long long;
using ld = long double;

int main() {
    string s;
    cin >> s;
    if (s == "first") {
        int n;
        cin >> n;
        vector<int> a(n);
        ll sum = 0;
        for (int& i : a) {
            cin >> i;
            sum += i;
        }
        cout << sum * (ll)1e8 << endl;
        cout.flush();
    } else {
        int n;
        ll x;
        cin >> n;
        ll res = 0;
        for (int i = 0; i < n; ++i) {
            cin >> x;
            res += x;
        }
        ll as = res / (ll)1e8;
        as /= n;
        cout << as + res % (ll)1e8 << endl;
    }
    return 0;
}
```

Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>
#include <map>
#include <unordered_map>
#include <random>
#include <chrono>

using namespace std;
using ll = long long;
using ld = long double;

using pii = pair<int, int>;

set<pii> v;

void solveFor2(pii a, pii b) {
    int x1 = a.second, x2 = b.second, y1 = a.first, y2 = b.first;
    if (x1 == x2) {
        int x = x1;
        int y = y2 - y1;
        if (x > y) {
            swap(x, y);
            v.emplace(x, y);
        }
    }
    if (x1 == y2) {
        int y = y2;
        int x = x2 - y1;
        if (x > y) {
            swap(x, y);
            v.emplace(x, y);
        }
    }
}

bool checkRect(vector<vector<int>>& f, int x, int y, int xS, int yS, int k) {
    int ch = 1;
    for (int i = y; i < y + yS; ++i) {
        for (int j = x; j < x + xS; ++j) {
            if (f[i][j] == k) {
                ch = 0;
                return ch;
            }
        }
    }
    return ch;
}

void findRect(vector<vector<int>>& f) {
    for (vector<int> i : f) {
        for (int j : i) cout << j << " ";
        cout << endl;
    }
    cout << "/////////////////////////////////////" << endl;
    int ch = 1;
    int len = 0;
    int y = 0;
    for (int i = 0; i < f.size(); ++i) {
        ++y;
        if (!ch) {
            break;
        }
        if (f[i][0] == 1) {
            if (i == 0) {
                ch = 0;
            } else {
                int curLen = 0;
                for (int j = 0; j < f[0].size(); ++j) {
```

```

        if (f[i][j] == 1) curLen++;
        else {
            break;
        }
    }
    if (curLen != len) {
        ch = 0;
    }
}
}
int curLen = 0;
for (int j = 0; j < f[0].size(); ++j) {
    if (f[i][j] == 0) {
        curLen++;
    } else {
        if (i == 0) {
            len = curLen;
        } else {
            if (curLen != len) {
                ch = 0;
                break;
            }
        }
    }
}
}
if (ch) {
    if (y > len) {
        swap(y, len);
    }
    v.emplace(y, len);
}
}

void solveFor3(pii a, pii b, pii c) {
    // a < b < c

    if (c.first > a.first && a.first + b.first > c.first && a.first <= c.first && b.first <= c.first && a.second + b.second <= c.second) {
        v.emplace(min(a.second, c.first - a.first), max(a.second, c.first - a.first));
    }
    swap(a.first, a.second);
    if (c.first > a.first && a.first + b.first > c.first && a.first <= c.first && b.first <= c.first && a.second + b.second <= c.second) {
        v.emplace(min(a.second, c.first - a.first), max(a.second, c.first - a.first));
    }
    swap(b.first, b.second);

    if (c.first > a.first && a.first + b.first > c.first && a.first <= c.first && b.first <= c.first && a.second + b.second <= c.second) {
        v.emplace(min(a.second, c.first - a.first), max(a.second, c.first - a.first));
    }
    swap(a.first, a.second);

    if (c.first > a.first && a.first + b.first > c.first && a.first <= c.first && b.first <= c.first && a.second + b.second <= c.second) {
        v.emplace(min(a.second, c.first - a.first), max(a.second, c.first - a.first));
    }
}

void solveW(pii a, pii b, pii c) {
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
    swap(a.first, a.second);
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
    swap(a.first, a.second);
    swap(b.first, b.second);
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
    swap(a.first, a.second);
}

```

```

    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
}

void solveH(pii a, pii b, pii c) {
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
    swap(a.first, a.second);
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
    swap(a.first, a.second);
    swap(b.first, b.second);
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
    swap(a.first, a.second);
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
}

int main() {
    vector<pair<int, int>> rect;
    for (int i = 0; i < 3; ++i) {
        int x, y;
        cin >> x >> y;
        if (x > y) {
            swap(x, y);
        }
        rect.emplace_back(x, y);
    }
    sort(rect.begin(), rect.end());
    solveFor2(rect[0], rect[1]);
    solveFor2(rect[0], rect[2]);
    solveFor2(rect[1], rect[2]);
    v.insert(rect[0]);
    v.insert(rect[1]);
    v.insert(rect[2]);
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 2; ++j) {
            if (i != j)
                solveFor3(rect[i], rect[j], rect[2]);
        }
    }
    solveW(rect[0], rect[1], rect[2]);
    solveW(rect[1], rect[0], rect[2]);
    solveH(rect[0], rect[1], rect[2]);
    solveH(rect[1], rect[0], rect[2]);
    for (auto i : v) {
        int f = i.first;
        int s = i.second;
        if (f > s) swap(f, s);
        if (f > 0 && s > 0) {
            cout << f << " " << s << endl;
        }
    }
    return 0;
}

```

Task D ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>
#include <map>
#include <unordered_map>
#include <random>
#include <chrono>

using namespace std;
using ll = long long;
using ld = long double;

struct Heap {
    int sz;
    int startSz;
    bool reload;
    Heap(int _sz) {
        sz = _sz;
        startSz = _sz;
        reload = false;
    }
};

map<vector<int>, int> calc;

int solve(Heap h) {
    if (calc.find({h.sz, h.startSz, h.reload}) != calc.end()) {
        return calc[{h.sz, h.startSz, h.reload}];
    }
    if (h.sz == 0 && h.reload) {
        return 0;
    }
    vector<int> moves;
    for (int i = 1; i <= h.sz; ++i) {
        Heap h1 = h;
        h1.sz -= i;
        moves.emplace_back(solve(h1));
    }
    if (!h.reload) {
        Heap h1 = h;
        h1.sz = h1.startSz;
        h1.reload = true;
        moves.emplace_back(solve(h1));
    }
    sort(moves.begin(), moves.end());
    int cur = 0;
    for (int i : moves) {
        if (i == cur) {
            ++cur;
        } else if (i > cur) {
            break;
        }
    }
    calc[{h.sz, h.startSz, h.reload}] = cur;
    return cur;
}

void play(vector<int>& g, vector<Heap> h) {
    int num1 = 0, num2 = 0;
    while (true) {
        int x = 0;
        for (int i : g) {
            x ^= i;
        }
        int bit = 0;
        for (int i = 0; i < 64; ++i) {
            if ((x >> i) & 1) {
                bit = i;
            }
        }
    }
}
```

```

    }
    int move = 0;
    for (int i = 0; i < g.size(); ++i) {
        if (move) break;
        if ((g[i] >> bit) & 1) {
            for (int j = 1; j <= h[i].sz; ++j) {
                if (calc[{h[i].sz - j, h[i].startSz, h[i].reload}] == (g[i] ^ x)) {
                    cout << i + 1 << "┘" << j << endl;
                    move = 1;
                    h[i].sz -= j;
                    g[i] = g[i] ^ x;
                }
            }
            if (!move && !h[i].reload) {
                if (calc[{h[i].startSz, h[i].startSz, !h[i].reload}] == (g[i] ^ x)) {
                    h[i].sz = h[i].startSz;
                    h[i].reload = true;
                    g[i] = g[i] ^ x;
                    move = 1;
                    cout << i + 1 << "┘" << 0 << endl;
                }
            }
        }
    }

    cin >> num1 >> num2;
    if (num1 == -1) {
        break;
    }
    --num1;
    if (num2 != 0) {
        h[num1].sz -= num2;
        g[num1] = calc[{h[num1].sz, h[num1].startSz, h[num1].reload}];
    } else {
        h[num1].sz = h[num1].startSz;
        h[num1].reload = true;
        g[num1] = calc[{h[num1].sz, h[num1].startSz, h[num1].reload}];
    }
}

}

int main() {
    int n, x;
    cin >> n;
    vector<Heap> a;
    for (int i = 0; i < n; ++i) {
        cin >> x;
        a.emplace_back(x);
    }
    vector<int> g;
    for (Heap i : a) {
        g.emplace_back(solve(i));
    }
    x = 0;
    for (int i : g) {
        x ^= i;
    }
    if (x == 0) {
        cout << -1 << "┘" << -1 << endl;
        cout.flush();
    } else {
        play(g, a);
    }
    return 0;
}

```

Task E ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>
#include <map>
#include <unordered_map>
#include <random>
#include <chrono>

using namespace std;
using ll = long long;
using ld = long double;

using pii = pair<int, int>;

set<pii> v;

void solveFor2(pii a, pii b) {
    int x1 = a.second, x2 = b.second, y1 = a.first, y2 = b.first;
    if (x1 <= x2 && y1 <= y2) {
        if (x1 == x2) {
            for (int i = 1; i <= y2 - y1; ++i) {
                int x = x1;
                int y = i;
                if (x > y) {
                    swap(x, y);
                }
                v.emplace(x, y);
            }
        }
        if (x1 == y2) {
            for (int i = 1; i <= x2 - y1; ++i) {
                int x = y2;
                int y = i;
                if (x > y) {
                    swap(x, y);
                }
                v.emplace(x, y);
            }
        }
    }
}

bool checkRect(vector<vector<int>>& f, int x, int y, int xS, int yS, int k) {
    int ch = 1;
    for (int i = y; i < y + yS; ++i) {
        for (int j = x; j < x + xS; ++j) {
            if (f[i][j] == k) {
                ch = 0;
                return ch;
            }
        }
    }
    return ch;
}

void findRect(vector<vector<int>>& f) {
    for (vector<int> i : f) {
        for (int j : i) cout << j << " ";
        cout << endl;
    }
    cout << "////////////////////////////////////" << endl;
    int ch = 1;
    int len = 0;
    int y = 0;
    for (int i = 0; i < f.size(); ++i) {
        ++y;
        if (!ch) {
            break;
        }
    }
}
```



```

        if (f[i][0] == 1) {
            if (i == 0) {
                ch = 0;
            } else {
                int curLen = 0;
                for (int j = 0; j < f[0].size(); ++j) {
                    if (f[i][j] == 1) curLen++;
                    else {
                        break;
                    }
                }
                if (curLen != len) {
                    ch = 0;
                }
            }
        }
        int curLen = 0;
        for (int j = 0; j < f[0].size(); ++j) {
            if (f[i][j] == 0) {
                curLen++;
            } else {
                if (i == 0) {
                    len = curLen;
                } else {
                    if (curLen != len) {
                        ch = 0;
                        break;
                    }
                }
            }
        }
    }
}
if (ch) {
    if (y > len) {
        swap(y, len);
    }
    v.emplace(y, len);
}
}

void solveFor3(pii a, pii b, pii c) {
    // a < b < c
    if (a.first + b.first > c.first && a.first <= c.first && b.first <= c.first && a.second + b.
        second <= c.second) {
        v.emplace(min(a.second, c.first - a.first), max(a.second, c.first - a.first));
    }
    swap(a.first, a.second);
    if (a.first + b.first > c.first && a.first <= c.first && b.first <= c.first && a.second + b.
        second <= c.second) {
        v.emplace(min(a.second, c.first - a.first), max(a.second, c.first - a.first));
    }
}

void solveW(pii a, pii b, pii c) {
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
    swap(a.first, a.second);
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
    swap(a.first, a.second);
    swap(b.first, b.second);
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
    swap(a.first, a.second);
    if (a.first == b.first && a.second + b.second == c.second) {
        v.emplace(min(c.first - b.first, c.second), max(c.first - b.first, c.second));
    }
}

void solveH(pii a, pii b, pii c) {
    if (a.first + b.first == c.first && a.second == b.second) {

```

```

        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
    swap(a.first, a.second);
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
    swap(a.first, a.second);
    swap(b.first, b.second);
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
    swap(a.first, a.second);
    if (a.first + b.first == c.first && a.second == b.second) {
        v.emplace(min(c.second - b.second, c.first), max(c.second - b.second, c.first));
    }
}

int main() {
    int n;
    cin >> n;
    string s;
    cin >> s;
    if (s == "transmit") {
        int x;
        for (int i = 0; i < n; ++i) {
            cin >> x;
            int cur = 0;
            for (int i = 0; i < 10; ++i) {
                string s1 = "";
                for (int j = 0; j < 10; ++j) {
                    if (cur < x) {
                        s1 += '1';
                        cur++;
                    } else {
                        s1 += '0';
                    }
                }
                cout << s1 << endl;
            }
            cout << endl;
        }
    } else {
        string k;
        for (int h = 0; h < n; ++h) {
            int cur = 0;
            for (int i = 0; i < 10; ++i) {
                cin >> k;
                for (char c : k) {
                    if (c == '1') cur++;
                }
            }
            cout << cur << endl;
        }
    }
    return 0;
}

```

Task F ()

```
a = input()
b = input()
ind = 0
f = ""
while ind < len(a):
    if a[ind] != '(':
        f += a[ind]
        ind += 1
    else:
        ind += 1
        cur = ""
        cnt = ""
        while a[ind] != '|':
            cur += a[ind]
            ind += 1
        ind += 1
        while a[ind] != ')':
            cnt += a[ind]
            ind += 1
        ind += 1
        f += cur * int(cnt)
s = ""
ind = 0
while ind < len(b):
    if b[ind] != '(':
        s += b[ind]
        ind += 1
    else:
        ind += 1
        cur = ""
        cnt = ""
        while b[ind] != '|':
            cur += b[ind]
            ind += 1
        ind += 1
        while b[ind] != ')':
            cnt += b[ind]
            ind += 1
        ind += 1
        s += cur * int(cnt)
print(int(f) + int(s))
```