

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	28	65	453

Task A ()

```
//#define _GLIBCXX_DEBUG
#include <bits/stdc++.h>
#pragma GCC optimize("Ofast")

using namespace std;

const long long K = 1150;
const long long MAXN = 1e5 + 1;
const long long C = 1000;
const long long MOD = 1e9 + 7;
const long long INF=1e9;

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    vector<int> ans;
    for (int i = 0; i < 6; i++) {
        int x;
        cin >> x;
        x--;
        ans.insert(ans.begin()+x, i+1);
    }
    vector<int> ans1(7, 0);
    for (int i = 0; i < 6; i++) {
        ans1[ans[i]] = i+1;
    }
    for (int i = 1; i <= 6; i++) {
        cout << ans1[i] << " ";
    }
}
```


Task B ()

```
//#define _GLIBCXX_DEBUG

#include <bits/stdc++.h>
//#pragma GCC optimize("Ofast")

using namespace std;

const long long K = 1150;
const long long MAXN = 1e5 + 1;
const long long C = 1000;
const long long MOD = 1e9;
const long long INF=1e9;

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    string s;
    cin >> s;
    if (s=="first") {
        long long n;
        cin >> n;
        vector<long long> a(n);
        long long x = 0;
        for (long long i = 0; i<n; i++) {
            cin >> a[i];
            x += a[i];
        }
        x *= (long long)MOD;
        cout << x << "\n";
    }
    else {
        long long n;
        cin >> n;
        vector<long long> a(n);
        long long x = 0;
        long long s= 0;
        for (long long i = 0; i<n; i++) {
            cin >> a[i];
            s += a[i]%(MOD);
            a[i] -= a[i]%MOD;
            x = a[i]/MOD;
        }
        cout << s + x << "\n";
    }
}
```


Task C ()

```
#define _GLIBCXX_DEBUG

#include <bits/stdc++.h>
// #pragma GCC optimize("Ofast")

using namespace std;

const long long K = 1150;
const long long MAXN = 1e5 + 1;
const long long C = 1000;
const long long MOD = 1e9;
const long long INF = 1e9;

bool check(int x, int y, int a, int b, vector<vector<int>>&h, vector<vector<int>>&v) {
    bool lr = false, tb = false;
    for (int i = x; i < x+a; i++) {
        if (v[i][y] || v[i][y+b]) {
            lr = true;
        }
    }
    for (int j = y; j < y+b; j++) {
        if (h[x][j] || h[x+a][j]) {
            tb = true;
        }
    }
    for (int i = x; i < x+a; i++) {
        for (int j = y+1; j < y+b; j++) {
            if (v[i][j]) {
                lr = false;
            }
        }
    }
    for (int i = x+1; i < x+a; i++) {
        for (int j = y; j < y+b; j++) {
            if (h[i][j]) {
                tb = false;
            }
        }
    }
    return lr && tb;
}

void add(int x, int y, int a, int b, vector<vector<int>>&h, vector<vector<int>>&v) {
    bool lr = false, tb = false;
    for (int i = x; i < x+a; i++) {
        v[i][y]++;
        v[i][y+b]++;
    }
    for (int j = y; j < y+b; j++) {
        h[x][j]++;
        h[x+a][j]++;
    }
}

int mxx = -1;
int mnx = 1e9;
int mxy = -1;
int mny = 1e9;
int mxxcnt = 0;
int mxycnt = 0;
int mnxcnt = 0;
int mnycnt = 0;
void dfs(int x, int y, vector<vector<int>>&cmp, int c, vector<vector<int>>&h, vector<vector<int>>&v) {
    cmp[x][y] = c;
    if (x > mxx) {
        mxxcnt = 0;
    }
    mxx = max(x, mxx);
    if (mxx == x) {
        mxxcnt++;
    }
}
```



```

    }
    if (x < mnx) {
        mnxcnt=0;
    }
    mnx=min(x, mnx);
    if (x==mnx) {
        mnxcnt++;
    }
    if (y > mxy) {
        mxycnt=0;
    }
    mxy = max(y, mxy);
    if (y==mxy) {
        mxycnt++;
    }
    if (y < mny) {
        mnycnt=0;
    }
    mny=min(y, mny);
    if (y==mny) {
        mnycnt++;
    }
    if (!h[x+1][y] && cmp[x+1][y]==-1) {
        dfs(x+1, y, cmp, c, h, v);
    }
    if (!h[x][y] && cmp[x-1][y]==-1) {
        dfs(x-1, y, cmp, c, h, v);
    }
    if (!v[x][y+1] && cmp[x][y+1] == -1) {
        dfs(x, y+1, cmp, c, h, v);
    }
    if (!v[x][y] && cmp[x][y-1] == -1) {
        dfs(x, y-1, cmp, c, h, v);
    }
}

}

signed main(){
    vector<pair<int, int>> pa(3);
    set<pair<int, int>> ans;
    for (int i = 0; i < 3; i++) {
        cin >> pa[i].first >> pa[i].second;
        ans.insert({min(pa[i].first, pa[i].second), max(pa[i].first, pa[i].second)});
    }
    sort(pa.begin(), pa.end());
    do {
        for (int mask = 0; mask < (1<<3); mask++) {
            vector<pair<int, int>> a= pa;
            for (int i = 0; i < 3; i++) {
                if ((mask >> i)&1) {
                    swap(a[i].first, a[i].second);
                }
            }
            int n = a[0].first;
            int m = a[0].second;
            for (int x1 = 0; x1+a[1].first<=n; x1++) {
                for (int y1 = 0; y1 + a[1].second <= m; y1++) {
                    {
                        vector<vector<int>> vert(n, vector<int>(m+1, 0));
                        vector<vector<int>> horiz(n+1, vector<int>(m, 0));
                        for (int i = 0; i < n; i++) {
                            vert[i][0] = 1;
                            vert[i][m]=1;
                        }
                        for (int i = 0; i < m; i++) {
                            horiz[0][i] = 1;
                            horiz[n][i] = 1;
                        }
                        if (!check(x1, y1, a[1].first, a[1].second, horiz, vert)) {
                            continue;
                        }
                    }
                    add(x1, y1, a[1].first, a[1].second, horiz, vert);
                }
            }
        }
    }
}

```



```

    }
}
while(next_permutation(pa.begin(), pa.end()));
for (auto x:ans) {
    cout << x.first << " " << x.second << "\n";
}
}

```


Task D ()

```
// #define _GLIBCXX_DEBUG

#include <bits/stdc++.h>
// #pragma GCC optimize("Ofast")

using namespace std;

const long long K = 1150;
const long long MAXN = 1e5 + 1;
const long long C = 1000;
const long long MOD = 1e9;
const long long INF = 1e9;

signed main() {
    int n;
    cin >> n;
    vector<int> a(n);
    vector<int> start(n);
    int s = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        start[i] = a[i];
        s ^= (a[i] + 1);
    }
    if (s == 0) {
        cout << "-1_-1" << endl;
        return 0;
    }

    vector<bool> used(n, false);
    while (true) {
        for (int i = 0; i < n; i++) {
            int nach = 0;
            if (!used[i] && a[i] == start[i]) {
                nach = start[i] + 1;
            }
            else {
                nach = a[i];
            }
            if (!used[i] && (nach ^ start[i]) == s) {
                used[i] = true;
                a[i] = start[i];
                cout << i + 1 << "_ " << 0 << endl;
                s = 0;
                break;
            }
            else {
                bool get = false;
                for (int j = 0; j < a[i]; j++) {
                    if ((nach ^ j) == s) {

                        cout << i + 1 << "_ " << a[i] - j << endl;
                        a[i] = j;
                        s = 0;
                        get = true;
                        break;
                    }
                }
                if (get) {
                    break;
                }
            }
        }
    }
    int x, cnt;
    cin >> x >> cnt;
    if (x == -1) {
        return 0;
    }
    x--;
    if (cnt == 0) {
        used[x] = true;
    }
}
```



```

        a[x] = start[x];
    }
    else {
        a[x] -= cnt;
    }
    s = 0;
    for (int i = 0; i < n; i++) {
        if (!used[i] && a[i] == start[i]) {
            s ^= start[i] + 1;
        }
        else {
            s ^= a[i];
        }
    }
}

}

```


Task E ()

```
#define _GLIBCXX_DEBUG

#include <bits/stdc++.h>
//#pragma GCC optimize("Ofast")

using namespace std;

const long long K = 1150;
const long long MAXN = 1e5 + 1;
const long long C = 1000;
const long long MOD = 1e9;
const long long INF=1e9;

signed main(){
    int t;
    cin >> t;
    string s;
    cin >> s;
    while(t--){
        if(s=="transmit") {
            int n;
            cin >> n;
            vector<string>ans(10);
            for(int i = 0;i<10;i++) {
                for (int j = 0;j<10;j++) {
                    ans[i].push_back('0');
                }
            }
            for (int k = 0;k<n;k++) {
                ans[k/10][k%10]='1';
            }
            for (int i = 0;i<10;i++) {
                cout << ans[i] << "\n";
            }
        }
        else {
            vector<string>ans(10);
            int cnt=0;
            for (int i = 0;i<10;i++) {
                cin >> ans[i];
                for (auto c:ans[i]) {
                    if (c=='1') {
                        cnt++;
                    }
                }
            }
            cout << cnt << "\n";
        }
    }
}
```


Task F ()

```
#define _GLIBCXX_DEBUG

#include <bits/stdc++.h>
//#pragma GCC optimize("Ofast")

using namespace std;

const long long K = 1150;
const long long MAXN = 1e5 + 1;
const long long C = 1000;
const long long MOD = 1e9;
const long long INF=1e9;

signed main(){
    string sa,sb;
    cin >> sa >> sb;
    vector<pair<int, int>> ans;
    vector<pair<int, int>> a;
    vector<pair<int, int>> b;
    for (int i = 0;i<sa.size();i++) {
        if (sa[i] >= '0' && sa[i] <= '9') {
            a.push_back({1, sa[i]-'0'});
        }
        else {
            int d = sa[i+1] - '0';
            int cnt =0 ;
            for (int j = i+3;sa[j] != ' ');j++) {
                i = j+1;
                cnt *= 10;
                cnt += sa[j] - '0';
            }
            a.push_back({cnt, d});
        }
    }
    for (int i = 0;i<sb.size();i++) {
        if (sb[i] >= '0' && sb[i] <= '9') {
            b.push_back({1, sb[i]-'0'});
        }
        else {
            int d = sb[i+1] - '0';
            int cnt =0 ;
            for (int j = i+3;sb[j] != ' ');j++) {
                i = j+1;
                cnt *= 10;
                cnt += sb[j] - '0';
            }
            b.push_back({cnt, d});
        }
    }
    int x = a.size()-1;
    int y=b.size()-1;
    int d = 0;
    while(x>=0 || y>=0) {
        if (x >= 0 && y >= 0) {
            int sz = min(a[x].first, b[y].first);
            a[x].first -= sz;
            b[y].first -=sz;
            int last= a[x].second + b[y].second;
            if (d > 0) {
                last++;
                d=0;
            }
            ans.push_back({1, last%10});
            if (last >= 10) {
                ans.push_back({sz-1, (a[x].second + b[y].second + 1) % 10});
                d = 1;
            }
        }
        else {
            ans.push_back({sz-1, (a[x].second + b[y].second) % 10});
            d =0;
        }
    }
}
```



```

    }

}

else if (x>=0) {
    int sz=a[x].first;
    pair<int, int> f={sz, 0};
    a[x].first==sz;
    int last= a[x].second + f.second;
    if (d > 0) {
        last++;
        d=0;
    }
    ans.push_back({1, last%10});
    if (last >= 10) {
        ans.push_back({sz-1, (a[x].second + f.second + 1) % 10});
        d = 1;
    }
    else {
        ans.push_back({sz-1, (a[x].second + f.second) % 10});
        d =0;
    }
}

}

else if (y>=0) {
    int sz=b[y].first;
    pair<int, int> f={sz, 0};
    b[y].first==sz;
    int last= f.second + b[y].second;
    if (d > 0) {
        last++;
        d=0;
    }
    ans.push_back({1, last%10});
    if (last >= 10) {
        ans.push_back({sz-1, (f.second + b[y].second + 1) % 10});
        d = 1;
    }
    else {
        ans.push_back({sz-1, (f.second + b[y].second) % 10});
        d =0;
    }
}

}

while(x>=0&&a[x].first==0) {
    x--;
    a.pop_back();
}

while (y>=0&&b[y].first==0) {
    y--;
    b.pop_back();
}

}

if (d>0) {
    ans.push_back({1, 1});
}

while(ans.back().second == 0) {
    ans.pop_back();
}

reverse(ans.begin(), ans.end());
for (auto c:ans) {
    if (c.first!= 0){
        cout << "(" << c.second << "|" << c.first << ")";
    }
}

}

}

```