Олимпиада СПбГУ по информатике 2018/19 учебного года

Лучинин Алексей Михайлович

A	B	C	D	E	F	Sum
100	100	100	60	35	24	419

Task A (100)

```cpp
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>
#include<iomanip>
#include<algorithm>
#include<cmath>

using namespace std;

#define int long long

signed main() {
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);

        int n, m;
        cin >> n >> m;

        if (m % n == 0) {
                int a = m / n;
                while (a % 2 == 0)
                        a /= 2;
                if (a == 1)
                        cout << "Yes";
                else
                        cout << "No";
        }
        else
                cout << "No";

        return 0;
}
```

Task B (100)

```cpp
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>
#include<iomanip>
#include<algorithm>
#include<cmath>

using namespace std;

#define int long long

signed main() {
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);

        int n;
        cin >> n;
        string s;
        cin >> s;

        bool f = false;
        for (int i = 0; i < n - 1; i++) {
                if (s[i] == 'o' && s[i + 1] == 'r')
                        f = true;
        }

        for (int i = 0; i < n - 2; i++) {
                if (s[i] == 'o' && s[i + 2] == 'r')
                        f = true;
        }

        for (int i = 0; i < n - 1; i++) {
                if (s[i] == 'r' && s[i + 1] == 'o')
                        f = true;
        }

        if (f)
                cout << "Yes";
        else
                cout << "No";

        return 0;
}
```

Task C (100)

```
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>
#include<iomanip>
#include<algorithm>
#include<cmath>

using namespace std;

#define int long long

vector<bool> used;
vector<vector<int>> edges;
map<pair<int, int>, int> mp;
vector<int> c;
int n;

int dfs(int v) {
        int cnt = 1;
        used[v] = true;
        for (int& x : edges[v]) {
                if (!used[x]) {
                        int d = dfs(x);
                        cnt += d;
                        mp[{v, x}] = d;
                        mp[{x, v}] = n - d;
                }
        }
        c[v] = cnt;
        return cnt;
}

signed main() {
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);

        cin >> n;
        used.resize(n);
        edges.resize(n);
        c.resize(n);

        for (int i = 0; i < n - 1; i++) {
                int a, b;
                cin >> a >> b;
                a--;
                b--;
                edges[a].push_back(b);
                edges[b].push_back(a);
        }

        dfs(0);
        for (int i = 0; i < n; i++) {
                int maxi = 1;
                for (int& x : edges[i]) {
                        maxi = max(maxi, mp[{i, x}] + 1);
                }
                cout << maxi << '␣';
        }

        return 0;
}
```

Task D (60)

```
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>
#include<iomanip>
#include<algorithm>
#include<cmath>

using namespace std;

#define int long long

signed main() {
	ios_base::sync_with_stdio(0);
	cin.tie(0);
	cout.tie(0);

	string q;
	cin >> q;
	int t, n, p;
	cin >> t >> n >> p;
	if (q == "split") {
		while (t--) {
			string s;
			cin >> s;
			if (n == 3) {
				cout << 'a' << s.substr(0, 6) << ' ';
				cout << 'b' << s.substr(0, 3) << s.substr(6, 3) << ' ';
				cout << 'c' << s.substr(3, 6);
			}
			else {
				cout << 'a' << s.substr(0, 6) << ' ';
				cout << 'a' << s.substr(0, 6) << ' ';
				cout << 'b' << s.substr(0, 3) << s.substr(6, 3) << ' ';
				cout << 'c' << s.substr(3, 6) << ' ';
				cout << 'c' << s.substr(3, 6);
			}
			cout << endl;
		}
	}
	else {
		while (t--) {
			vector<char> ans(9);
			for (int i = 0; i < (n + 1) / 2; i++) {
				string s;
				cin >> s;
				if (s[0] == 'a') {
					for (int i = 0; i < 6; i++)
						ans[i] = s[i + 1];
				}
				else if (s[0] == 'b') {
					for (int i = 0; i < 3; i++)
						ans[i] = s[i + 1];
					for (int i = 6; i < 9; i++)
						ans[i] = s[i - 2];
				}
				else {
					for (int i = 3; i < 9; i++)
						ans[i] = s[i - 2];
				}
			}
			for (char& c : ans)
				cout << c;
			cout << endl;
		}
	}

	return 0;
}
```

Task E (35)

```
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>
#include<iomanip>
#include<algorithm>
#include<cmath>

using namespace std;

#define int long long

const double inf = 2e9 + 9;
const double eps = 1e-12;

double len(double x1, double y1, double x2, double y2) {
        return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}

signed main() {
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);

        int n;
        cin >> n;

        vector<pair<double, double>> lst(n);
        for (int i = 0; i < n; i++) {
                cin >> lst[i].first >> lst[i].second;
        }

        double x1, y1;
        cin >> x1 >> y1;
        double x2, y2;
        cin >> x2 >> y2;

        double a = y2 - y1;
        double b = x1 - x2;
        double c = -x1 * y2 + x2 * y1;
        vector<double> temp(n);
        if (x1 != x2) {
                double k = -a / b;
                double d = -c / b;
                if (k == 0) {
                        if (x2 > x1) {
                                double maxi = lst[0].first;
                                int index = 0;
                                for (int i = 1; i < n; i++) {
                                        if (maxi < lst[i].first) {
                                                maxi = lst[i].first;
                                                index = i;
                                        }
                                }
                                int cnt = 0;
                                for (int i = 0; i < n; i++) {
                                        if (maxi == lst[i].first)
                                                cnt++;
                                }
                                if (cnt == 1)
                                        cout << index + 1;
                                else {
                                        cnt = 0;
                                        double delta = inf;
                                        for (int i = 0; i < n; i++) {
                                                if (maxi == lst[i].first) {
                                                        if (abs(a * lst[i].first + b * lst[i].
                                                            second + c) / (a * a + b * b) < delta)
```

```
                                             {
                                                index = i;
                                                delta = abs(a * lst[i].first + b *
                                                     lst[i].second + c) / (a * a +
                                                     b * b);
                                        }
                                }
                        }
                        for (int i = 0; i < n; i++) {
                                if (maxi == lst[i].first) {
                                        double de = abs(a * lst[i].first + b * lst
                                            [i].second + c) / (a * a + b * b);
                                        if (de == delta)
                                                cnt++;
                                }
                        }
                        if (cnt == 1)
                                cout << index + 1;
                        else
                                cout << -1;
                }
        }
        else {
                double maxi = lst[0].first;
                int index = 0;
                for (int i = 1; i < n; i++) {
                        if (maxi > lst[i].first) {
                                maxi = lst[i].first;
                                index = i;
                        }
                }
                int cnt = 0;
                for (int i = 0; i < n; i++) {
                        if (maxi == lst[i].first)
                                cnt++;
                }
                if (cnt == 1)
                        cout << index + 1;
                else {
                        cnt = 0;
                        double delta = inf;
                        for (int i = 0; i < n; i++) {
                                if (maxi == lst[i].first) {
                                        if (abs(a * lst[i].first + b * lst[i].
                                            second + c) / (a * a + b * b) < delta)
                                             {
                                                index = i;
                                                delta = abs(a * lst[i].first + b *
                                                     lst[i].second + c) / (a * a +
                                                     b * b);
                                        }
                                }
                        }
                        for (int i = 0; i < n; i++) {
                                if (maxi == lst[i].first) {
                                        double de = abs(a * lst[i].first + b * lst
                                            [i].second + c) / (a * a + b * b);
                                        if (de == delta)
                                                cnt++;
                                }
                        }
                        if (cnt == 1)
                                cout << index + 1;
                        else
                                cout << -1;
                }
        }
        return 0;
}
for (int i = 0; i < n; i++) {
        double X = lst[i].first;
        double Y = lst[i].second;
        double k1 = -1 / k;
        double del = Y - k1 * X;
        temp[i] = (del - a) / (k - k1);
}
if (x1 < x2) {
        double maxi = temp[0];
        int index = 0;
        for (int i = 1; i < n; i++) {
                if (maxi < temp[i]) {
                        maxi = temp[i];
                        index = i;
                }
        }
        int cnt = 0;
        for (int i = 0; i < n; i++) {
                if (maxi == temp[i])
                        cnt++;
        }
        if (cnt == 1)
                cout << index + 1;
        else {
                cnt = 0;
                double delta = inf;
                for (int i = 0; i < n; i++) {
                        if (maxi == temp[i]) {
                                if (abs(a * lst[i].first + b * lst[i].second + c)
                                    / (a * a + b * b) < delta) {
                                        index = i;
                                        delta = abs(a * lst[i].first + b * lst[i].
                                            second + c) / (a * a + b * b);
                                }
                        }
                }
                for (int i = 0; i < n; i++) {
                        if (maxi == temp[i]) {
                                double de = abs(a * lst[i].first + b * lst[i].
                                    second + c) / (a * a + b * b);
                                if (de == delta)
                                        cnt++;
                        }
                }
                if (cnt == 1)
                        cout << index + 1;
                else
                        cout << -1;
        }
}
else {
        double maxi = temp[0];
        int index = 0;
        for (int i = 1; i < n; i++) {
                if (maxi > temp[i]) {
                        maxi = temp[i];
                        index = i;
                }
        }
        int cnt = 0;
        for (int i = 0; i < n; i++) {
                if (maxi == temp[i])
                        cnt++;
        }
        if (cnt == 1)
                cout << index + 1;
        else {
                cnt = 0;
                double delta = inf;
                for (int i = 0; i < n; i++) {
                        if (maxi == temp[i]) {

                                if (abs(a * lst[i].first + b * lst[i].second + c)
                                    / (a * a + b * b) < delta) {
                                        index = i;
                                        delta = abs(a * lst[i].first + b * lst[i].
                                            second + c) / (a * a + b * b);
                                }
                        }
```

```
                    }
                    for (int i = 0; i < n; i++) {
                        if (maxi == temp[i]) {
                            double de = abs(a * lst[i].first + b * lst[i].
                              second + c) / (a * a + b * b);
                            if (de == delta)
                                cnt++;
                        }
                    }
                    if (cnt == 1)
                        cout << index + 1;
                    else
                        cout << -1;
                }
            }
        }
        else {
            if (x2 > x1) {
                double maxi = lst[0].second;
                int index = 0;
                for (int i = 1; i < n; i++) {
                    if (maxi < lst[i].second) {
                        maxi = lst[i].second;
                        index = i;
                    }
                }
                int cnt = 0;
                for (int i = 0; i < n; i++) {
                    if (maxi == lst[i].second)
                        cnt++;
                }
                if (cnt == 1)
                    cout << index + 1;
                else {
                    cnt = 0;
                    double delta = inf;
                    for (int i = 0; i < n; i++) {
                        if (maxi == lst[i].second) {
                            if (abs(a * lst[i].first + b * lst[i].second + c)
                              / (a * a + b * b) < delta) {
                                index = i;
                                delta = abs(a * lst[i].first + b * lst[i].
                                  second + c) / (a * a + b * b);
                            }
                        }
                    }
                    for (int i = 0; i < n; i++) {
                        if (maxi == lst[i].second) {
                            double de = abs(a * lst[i].first + b * lst[i].
                              second + c) / (a * a + b * b);
                            if (de == delta)
                                cnt++;
                        }
                    }
                    if (cnt == 1)
                        cout << index + 1;
                    else
                        cout << -1;
                }
            }
            else {
                double maxi = lst[0].second;
                int index = 0;
                for (int i = 1; i < n; i++) {
                    if (maxi > lst[i].second) {
                        maxi = lst[i].second;
                        index = i;
                    }
                }
                int cnt = 0;
                for (int i = 0; i < n; i++) {
                    if (maxi == lst[i].second)
                        cnt++;
                }
```

```
			if (cnt == 1)
				cout << index + 1;
			else {
				cnt = 0;
				double delta = inf;
				for (int i = 0; i < n; i++) {
					if (maxi == lst[i].second) {
						if (abs(a * lst[i].first + b * lst[i].second + c)
						    / (a * a + b * b) < delta) {
							index = i;
							delta = abs(a * lst[i].first + b * lst[i].
							    second + c) / (a * a + b * b);
						}
					}
				}
				for (int i = 0; i < n; i++) {
					if (maxi == lst[i].second) {
						double de = abs(a * lst[i].first + b * lst[i].
						    second + c) / (a * a + b * b);
						if (de == delta)
							cnt++;
					}
				}
				if (cnt == 1)
					cout << index + 1;
				else
					cout << -1;
			}
		}
	}


	return 0;
}
```

Task F (24)

```
#include<iostream>
#include<vector>
#include<set>
#include<map>
#include<string>
#include<iomanip>
#include<algorithm>
#include<cmath>

using namespace std;

#define int long long

//const int maxn = 1009;
//int dp[maxn][maxn][2];
//int lst[maxn][maxn][2];


int n, k, ans = 0;
vector<pair<int, int>> lst;
map<vector<int>, int> mp;

int xxx(int cnt, int a, int sa, int sb) {
        if (cnt == n) {
                ans = max(ans, a);
                return a;
        }
        vector<int> t{ cnt, a, sa, sb };
        if (mp[t] != 0)
                return mp[t];
        int x = lst[cnt].first;
        int y = lst[cnt].second;
        int q = xxx(cnt + 1, a + min(sa + k, x) + min(sb, y), max(0LL, sa + k - x), max(0LL, sb -
            y));
        int q1 = xxx(cnt + 1, a + min(sa, x) + min(sb + k, y), max(0LL, sa - x), max(0LL, sb + k -
             y));
        mp[t] = max(q, q1);
}

signed main() {
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);

        cin >> n >> k;
        lst.resize(n);
        for (int i = 0; i < n; i++) {
                cin >> lst[i].first >> lst[i].second;
        }

        xxx(0, 0, 0, 0);
        cout << ans;
        /*int sa = 0, sb = 0;
        for (int i = 1; i <= n; i++) {
                int a, b;
                cin >> a >> b;
                sa += a;
                sb += b;
                for (int x = 1; x <= i; x++) {
                        dp[i][x][0] = max(dp[i - 1][x - 1][0] + lst[i - 1][x - 1][0]
                }
        }*/


        return 0;
}
```