Олимпиада СПбГУ по информатике 2018/19 учебного года

Постников Егор Александрович

A	B	C	D	E	F	Sum
100	100	100	60	21	24	405

Task A (100)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;


int main()
{
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        int a, b;
        cin >> a >> b;
        while (a < b)
                a = a * 2;
        if (a == b)
                cout << "Yes";
        else
                cout << "No";
    return 0;
}
```

Task B (100)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>

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

#define int ll
#define all(x) x.begin(), x.end()

signed main()
{
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        int n;
        cin >> n;
        string s;
        cin >> s;
        s += "###";
        for (int i = 0; i < s.size() - 3; i++)
        {
                if (s[i] == 'o' && s[i + 1] == 'r' ||
                        s[i] == 'r' && s[i + 1] == 'o' ||
                        s[i] == 'o' && s[i + 2] == 'r')
                {
                        cout << "Yes";
                        return 0;
                }


        }
        cout << "No";
    return 0;
}
```

Task C (100)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>

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

#define int ll
#define all(x) x.begin(), x.end()

int n;
vector <vector <int>> a;
vector <int> subTree;
vector <int> ans;

int countSubTree(int now, int last)
{
	for (int v : a[now])
	{
		if (v == last)
			continue;
		subTree[now] += countSubTree(v, now);
	}
	subTree[now]++;
	return subTree[now];
}

void findAnswer(int now, int last, int pushed)
{
	int sum = 1;
	int maxi = pushed;
	for (int v : a[now])
	{
		if (v != last)
		{
			sum += subTree[v];
			maxi = max(maxi, subTree[v]);
		}
	}
	ans[now] = maxi + 1;
	for (int v : a[now])
	{
		if (v == last)
			continue;
		findAnswer(v, now, pushed + sum - subTree[v]);
	}
}

signed main()
{
#ifdef _DEBUG
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif
	int n;
	cin >> n;
	a.resize(n);
	subTree.resize(n);
	ans.resize(n);

	for (int i = 0; i < n - 1; i++)
	{
		int f, s;
		cin >> f >> s;
		f--;
		s--;
		a[f].push_back(s);
```

```
                a[s].push_back(f);
        }
        countSubTree(0, -1);
        findAnswer(0, -1, 0);

        for (int i = 0; i < n; i++)
        {
                cout << ans[i] << " ";
        }
    return 0;
}
```

Task D (60)

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <map>

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

#define int ll
#define all(x) x.begin(), x.end()

const ll INF = 1e9;

struct point
{
        int x, y;
        int num;
        void scan(int _num)
        {
                cin >> x >> y;
                num = _num;
        }

        int len2Int()
        {
                int nx = int(x);
                int ny = int(y);
                return nx * nx + ny * ny;
        }

        void remake()
        {
                while (abs(x) * 2 <= INF && abs(y) * 2 <= INF)
                {
                        x = x * 2;
                        y = y * 2;
                }
        }

        point() {}

        point(int _x, int _y)
        {
                x = _x;
                y = _y;
        }

        point operator -(const point &other)const
        {
                return point(x - other.x, y - other.y);
        }
        point operator +(const point &other)const
        {
                return point(x + other.x, y + other.y);
        }

        int operator * (const point &other)const
        {
                return x * other.y - y * other.x;
        }


};


bool cmpByX(point a, point b)
{
```

```
        if (a.x != b.x)
                return a.x < b.x;
        return a.y < b.y;
}

point zzz;
bool cmpByAngle(point a, point b)
{
        a = a - zzz;
        b = b - zzz;
        if (a * b == 0)
                return a.len2Int() < b.len2Int();
        return a * b > 0;
}

int sign(int a)
{
        if (a > 0)
                return 1;
        else if (a == 0)
                return 0;
        else
                return -1;
}

bool check(point a, point d, point c, point b)
{
        point ab = b - a;
        point ad = d - a;
        point ac = c - a;
        bool adBetweenCb = sign(ab * ad) * sign(ac * ad) <= 0;

        point ba = a - b;
        point bd = d - b;
        point bc = c - b;
        bool cbBetweenAd = sign(ba * bc) * sign(bd * bc) <= 0;

        return adBetweenCb * cbBetweenAd;
}

signed main()
{
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.precision(6);
        string s;
        cin >> s;
        if (s == "split")
        {
                int t, n, p;
                cin >> t >> n >> p;
                if (n == 3)
                {
                        for (int i = 0; i < t; i++)
                        {
                                string k;
                                cin >> k;
                                cout << k.substr(0, 6) + "a " + k.substr(3) + "b " + k.substr(0,
                                    3) + k.substr(6) + "c" << endl;
                        }
                }
                else if (n == 5)
                {
                        for (int i = 0; i < t; i++)
                        {
                                string k;
                                cin >> k;
                                string first = k.substr(0, 6) + "a ";
                                string second = k.substr(3) + "b ";
                                string third = k.substr(0, 3) + k.substr(6) + "c";
```

```
                                cout << first <<" "<< first <<" "<< second<<" "<<second <<" "<<
                                    third << endl;
                        }
                }
        }
        else
        {
                int t, n, p;
                cin >> t >> n >> p;
                if (n == 3 || n == 5)
                {
                        for (int i = 0; i < t; i++)
                        {
                                vector <pair<int, string>> kk((n + 1) / 2);
                                for (int i = 0; i < (n + 1) / 2; i++)
                                {
                                        cin >> kk[i].second;
                                        kk[i].first = kk[i].second.back() - 'a';
                                        kk[i].second.pop_back();
                                }
                                sort(all(kk));
                                unique(all(kk));

                                string f = kk[0].second;
                                string s = kk[1].second;
                                vector <int> ch{ kk[0].first, kk[1].first };

                                if (ch[0] == 0)
                                {
                                        if (ch[1] == 1)
                                        {
                                                cout << f + s.substr(3) << endl;
                                        }
                                        else
                                        {
                                                cout << f + s.substr(3) << endl;
                                        }
                                }
                                else
                                {
                                        cout << s.substr(0, 3) + f << endl;
                                }
                        }
                }
        }
    return 0;
}
```

Task E (21)

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <map>

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

#define int ll
#define all(x) x.begin(), x.end()

const ll INF = 1e9;

struct point
{
        int x, y;
        int num;
        void scan(int _num)
        {
                cin >> x >> y;
                num = _num;
        }

        int len2Int()
        {
                int nx = int(x);
                int ny = int(y);
                return nx * nx + ny * ny;
        }

        void remake()
        {
                while (abs(x) * 2 < INF && abs(y) * 2 < INF)
                {
                        x = x * 2;
                        y = y * 2;
                }
        }

        point() {}

        point(int _x, int _y)
        {
                x = _x;
                y = _y;
        }

        point operator -(const point &other)const
        {
                return point(x - other.x, y - other.y);
        }
        point operator +(const point &other)const
        {
                return point(x + other.x, y + other.y);
        }

        int operator * (const point &other)const
        {
                return x * other.y - y * other.x;
        }


};


bool cmpByX(point a, point b)
```

```
{
        if (a.x != b.x)
                return a.x < b.x;
        return a.y < b.y;
}

point zzz;
bool cmpByAngle(point a, point b)
{
        a = a - zzz;
        b = b - zzz;
        if (a * b == 0)
                return a.len2Int() < b.len2Int();
        return a * b > 0;
}

int sign(int a)
{
        if (a > 0)
                return 1;
        else if (a == 0)
                return 0;
        else
                return -1;
}

bool check(point a, point d, point c, point b)
{
        point ab = b - a;
        point ad = d - a;
        point ac = c - a;
        bool adBetweenCb = sign(ab * ad) * sign(ac * ad) <= 0;

        point ba = a - b;
        point bd = d - b;
        point bc = c - b;
        bool cbBetweenAd = sign(ba * bc) * sign(bd * bc) <= 0;

        return adBetweenCb * cbBetweenAd;
}

signed main()
{
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.precision(6);
        int n;
        cin >> n;
        vector <point> a(n);
        for (int i = 0; i < n; i++)
        {
                a[i].scan(i + 1);
        }
        point f, s;
        f.scan(-1);
        s.scan(-1);

        s = s - f;
        s.remake();
        s = f + s;


        zzz = a[0];
        int kk = 0;
        for (int i = 1; i < n; i++)
        {
                if (a[i].x < a[kk].x || a[i].x == a[kk].x && a[i].y < a[kk].y)
                        kk = i;
        }
        swap(a[0], a[kk]);
```

```
        zzz = a[kk];
        sort(a.begin() + 1, a.end(), cmpByAngle);

        vector <point> z;
        for (int i = 0; i < n; i++)
        {
                while (z.size() > 1 && sign((a[i] - z[z.size() - 1]) * (z[z.size() - 1] - z[z.size
                    () - 2])) > 0)
                {
                        z.pop_back();
                }
                z.push_back(a[i]);
        }
        a = z;
        n = z.size();

        map<int, vector<int>> cnt;
        for (int i = 0; i < n; i++)
        {
                int curLen = (a[i] - s).len2Int();
                cnt[curLen].push_back(a[i].num);
        }
        int ans;
        if (cnt.begin()->second.size() > 1)
                ans = -1;
        else
                ans = cnt.begin()->second[0];
        cout << ans;
        return 0;
}
```

Task F (24)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <map>

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

//#define int ll
#define all(x) x.begin(), x.end()

const ll INF = 1e9;

signed main()
{
#ifdef _DEBUG
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif
	ios_base::sync_with_stdio(0);
	cin.tie(0);
	cout.precision(6);
	int n, k;
	cin >> n >> k;
	vector < vector <vector <int>>> dp(35, vector <vector <int>>(1000, vector <int>(1000, -INF
	    )));
	dp[0][0][0] = 0;
	int maxi = 0;
	for (int i = 1; i <= n; i++)
	{
		int f, s;
		cin >> f >> s;
		for (int j = 0; j < 1000 - k; j++)
		{
			for (int z = 0; z < 1000 - k; z++)
			{
				if (dp[i - 1][j][z] < 0)
					continue;
				int profitF = min(j + k, f);
				int profitS = min(z, s);
				int fd = max(0, j + k - f);
				int sd = max(0, z - s);
				dp[i][fd][sd] = max(dp[i][fd][sd], dp[i - 1][j][z] + profitF +
				    profitS);

				profitF = min(j, f);
				profitS = min(z + k, s);
				fd = max(0, j - f);
				sd = max(0, z + k - s);
				dp[i][fd][sd] = max(dp[i][fd][sd], dp[i - 1][j][z] + profitF +
				    profitS);
			}
		}
	}
	for (int j = 0; j < 1000; j++)
	{
		for (int z = 0; z < 1000; z++)
		{
			maxi = max(maxi, dp[n][j][z]);
		}
	}
	cout << maxi;
    return 0;
}
```