Олимпиада СПбГУ по информатике 2018/19 учебного года

Михельсон Маргарита Николаевна

A	B	C	D	E	F	Sum
100	100	100	60	100	24	484

Task A (100)

```
#include <iostream>
#include <set>

using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);

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

    int fl = 1;
    while (n < m) n *= 2;

    if (n == m) cout << "Yes\n";
    else cout << "No\n";
    return 0;
}


//bkrimuukve
```

Task B (100)

```
#include <iostream>
#include <set>

using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);

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

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

    for (int i = 0; i < n-1; i++)
    {
        if (s[i+1] == 'o' && s[i] == 'r') fl = 1;
        if (i > 0 && s[i-1] == 'o' && s[i+1] == 'r') fl  =1;
        if (i+2 < n && s[i] == 'o' && s[i+2] == 'r') fl  =1;
    }

    if (fl) cout << "Yes\n";
    else cout << "No\n";
    return 0;
}


//bkrimuukve
```

Task C (100)

```
#include <iostream>
#include <set>
#include <vector>

using namespace std;

const int MX = 1e5 + 10;
vector <int> g[MX];
int cnt[MX], ans[MX];
int n;

void dfs(int v, int pr)
{
    cnt[v] = 1;
    int mx = 0;
    for (int u : g[v])
    {
        if (u == pr) continue;
        dfs(u, v);
        mx = max(mx, cnt[u]);
        cnt[v] += cnt[u];
    }

    mx = max(mx, n-cnt[v]);
    ans[v] = mx + 1;
}


int main()
{
    ios_base::sync_with_stdio(false);

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

    dfs(0, -1);

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


//bkrimuukve
```

Task D (60)

```
#include <iostream>
#include <set>
#include <string>

using namespace std;


void split_1()
{
    string s;
    cin >> s;
    cout << "a";
    for (int i = 3; i < 9; i++) cout << s[i];
    cout << "␣";

    cout << "b";
    for (int i = 0; i < 3; i++) cout << s[i];
    for (int i = 6; i < 9; i++) cout << s[i];
    cout << "␣";

    cout << "c";
    for (int i = 0; i < 6; i++) cout << s[i];
    cout << "\n";
}


void split_2()
{
    string s;
    cin >> s;
    cout << "a";
    for (int i = 3; i < 9; i++) cout << s[i];
    cout << "␣";

    cout << "a";
    for (int i = 3; i < 9; i++) cout << s[i];
    cout << "␣";

    cout << "b";
    for (int i = 0; i < 3; i++) cout << s[i];
    for (int i = 6; i < 9; i++) cout << s[i];
    cout << "␣";

    cout << "c";
    for (int i = 0; i < 6; i++) cout << s[i];
    cout << "␣";

    cout << "c";
    for (int i = 0; i < 6; i++) cout << s[i];
    cout << "\n";
}


void merge_1()
{
    string s[2];
    cin >> s[0] >> s[1];
    char ans[9];

    for (int i = 0; i < 2; i++)
    {
        if (s[i][0] == 'a')
        {
            for (int j = 3; j < 9; j++) ans[j] = s[i][j+1-3];
        }
        if (s[i][0] == 'b')
        {
            for (int j = 0; j < 3; j++) ans[j] = s[i][j+1];
            for (int j = 6; j < 9; j++) ans[j] = s[i][j+1-3];
        }
        if (s[i][0] == 'c')
        {
            for (int j = 0; j < 6; j++) ans[j] = s[i][j+1];
        }
    }


    for (int i = 0; i < 9; i++) cout << ans[i];
    cout << "\n";

}


void merge_2()
{
    string s[3];
    cin >> s[0] >> s[1] >> s[2];
    char ans[9];

    for (int i = 0; i < 3; i++)
    {
        if (s[i][0] == 'a')
        {
            for (int j = 3; j < 9; j++) ans[j] = s[i][j+1-3];
        }
        if (s[i][0] == 'b')
        {
            for (int j = 0; j < 3; j++) ans[j] = s[i][j+1];
            for (int j = 6; j < 9; j++) ans[j] = s[i][j+1-3];
        }
        if (s[i][0] == 'c')
        {
            for (int j = 0; j < 6; j++) ans[j] = s[i][j+1];
        }
    }


    for (int i = 0; i < 9; i++) cout << ans[i];
    cout << "\n";

}


int main()
{
    ios_base::sync_with_stdio(false);

    string s;
    cin >> s;
    int t, n, p;
    cin >> t >> n >> p;

    if (s[0] == 's')
    {

        for (int i = 0; i < t; i++)
        {
            if (n == 3) split_1();
            else split_2();
        }
        return 0;
    }
    else
    {
        for (int i = 0; i < t; i++)
        {
            if (n == 3) merge_1();
            else merge_2();
        }
    }


    return 0;
```

```
}

//asword   bpasrd   cpasswo
//bkrimuukve
```

Task E (100)

```
#include <iostream>
#include <set>
#include <algorithm>

using namespace std;

struct point
{
    long long x, y, nn;
    long long operator*(point a) { return x*a.x + y*a.y; }
    long long operator%(point a) { return x*a.y - y*a.x; }
    point operator+(point a) { return {x+a.x, y+a.y, nn}; }
    point operator-(point a) { return {x-a.x, y-a.y, nn}; }
};

point q;

bool cmp(point a, point b)
{
    if ((a*q) == (b*q)) return abs(a%q) > abs(b%q);
    return (a*q) < (b*q);
}

point read()
{
    point a;
    cin >> a.x >> a.y;
    return a;
}

//long long len2(point a) { return a.x*a.x + a.y*a.y; }


int main()
{
    ios_base::sync_with_stdio(false);

    int n;
    cin >> n;
    point p[n];
    for (int i = 0; i  <n; i++)
    {
        p[i] = read();
    }

    point p0 = read();
    q = read();
    q = q-p0;

    for (int i = 0; i < n; i++)
    {
        p[i] = p[i] - p0;
        p[i].nn = i;
    }

    sort(p, p+n, cmp);

    //cout << (p[n-1]*q) << "\n";
    //cout << (p[n-2]*q) << "\n";
    if (n >= 2 && (p[n-1]*q) == (p[n-2]*q) && ((p[n-1]%q) + (p[n-2]%q) == 0))
    {
        cout << "-1\n";
        return 0;
    }
    cout << p[n-1].nn + 1 << "\n";
    return 0;
}
```

```
//bkrimuukve
```

Task F (24)

```cpp
#include <iostream>
#include <set>

using namespace std;

int dp[5000][5000];

int main()
{
    ios_base::sync_with_stdio(false);

    //cout << 15*15*15 << "\n";
    int n, k;
    cin >> n >> k;
    int a[n], b[n];
    int na = 0, nb = 0;
    int pra[n], prb[n];

    for (int i = 0; i <  n; i++)
    {
        cin >> a[i] >> b[i];

        if (na + a[i] > (i+1)*k) a[i] = (i+1)*k - na;
        if (nb + b[i] > (i+1)*k) b[i] = (i+1)*k - nb;

        pra[i] = a[i];
        prb[i] = b[i];
        if (i > 0)
        {
            pra[i] += pra[i-1];
            prb[i] += prb[i-1];
        }
        na += a[i];
        nb += b[i];
    }

    na++;
    nb++;
    //cout << na << " " << nb << endl;

    //for (int i = 0; i < n; i++) cout << pra[i] << " " << prb[i] << "\n";

    int res = 0;

    for (int i = 0; i < na; i++)
        for (int j = 0; j < nb; j++) dp[i][j] = 0;

    for (int i = 0; i < na; i++)
    {
        for (int j = 0; j < nb; j++)
        {
            //cout << i << " " << j << endl;
            res = max(res, dp[i][j]);
            int t = (dp[i][j]+k-1)/k;
            if (t >= n) continue;
            int i1 = min(max(i+k, pra[t]), na-1), j1 = min(max(j+k, prb[t]), nb-1);
            int i2 = min(max(i, pra[t]), na-1), j2 = min(max(j, prb[t]), nb-1);
            dp[i1][j2] = max(dp[i1][j2], dp[i][j] + min(k, na-1-i));
            dp[i2][j1] = max(dp[i2][j1], dp[i][j] + min(k, nb-1-j));
        }
    }


    cout << res << endl;
    return 0;
}


//bkrimuukve
```