

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	100	0	440

Task A ()

```
#ifdef _DEBUG
#define _CRT_SECURE_NO_WARNINGS
#endif
#include <iostream>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <algorithm>
#include <queue>
#include <iomanip>
#include <cmath>

using namespace std;

const double EPS = 1e-7;
bool Less(double a, double b)
{
    return a + EPS < b;
}
bool More(double a, double b)
{
    return a - EPS > b;
}
bool Eq(double a, double b)
{
    return !Less(a, b) && !More(a, b);
}

const int SZ = 1e5 + 10;

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif
    long long n;
    cin >> n;
    cout << n - 1;
}
```

Task B ()

```
#ifdef _DEBUG
#define _CRT_SECURE_NO_WARNINGS
#endif
#include <iostream>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <algorithm>
#include <queue>
#include <iomanip>
#include <cmath>

using namespace std;

const double EPS = 1e-7, PI = atan2(0, -1);
bool Less(double a, double b)
{
    return a + EPS < b;
}
bool More(double a, double b)
{
    return a - EPS > b;
}
bool Eq(double a, double b)
{
    return !Less(a, b) && !More(a, b);
}

struct Point
{
    double x, y;
    Point() {}
    Point(double x, double y) : x(x), y(y) {}

    bool operator < (Point p)
    {
        if (x != p.x)
            return x < p.x;
        return y < p.y;
    }

    Point operator + (Point p)
    {
        return Point(x + p.x, y + p.y);
    }

    Point operator - (Point p)
    {
        return Point(x - p.x, y - p.y);
    }

    Point operator / (double d)
    {
        return Point(x / d, y / d);
    }

    Point operator * (double d)
    {
        return Point(x * d, y * d);
    }
};

double getAng(Point p)
{
    double d = atan2(p.y, p.x);
    if (Less(d, 0))
        d += 2 * PI;
    return d;
}

Point mid = { 0, 0 };
```

```

bool comp(Point p1, Point p2)
{
    p1 = p1 - mid;
    p2 = p2 - mid;
    return Less(getAng(p1), getAng(p2));
}

double getD(Point p)
{
    return sqrt(p.x * p.x + p.y * p.y);
}

Point mir(Point p1, Point p2)
{
    return p2 * 2.0 - p1;
}

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif
    int n;
    cin >> n;
    vector<Point> pol(6);
    if (n == 6)
    {
        for (int i = 0; i < 6; i++)
        {
            cin >> pol[i].x >> pol[i].y;
            mid = mid + pol[i];
        }
        mid = mid / 6.0;
        sort(pol.begin(), pol.end(), comp);
        cout << fixed << setprecision(5) << pol[0].x << '\u' << pol[0].y << '\u' << '\n';
        cout << fixed << setprecision(5) << pol[2].x << '\u' << pol[2].y << '\u' << '\n';
        cout << fixed << setprecision(5) << pol[4].x << '\u' << pol[4].y << '\u' << '\n';
        return 0;
    }

    for (int i = 0; i < 3; i++)
    {
        cin >> pol[i].x >> pol[i].y;
        mid = mid + pol[i];
    }
    mid = mid / 3.0;

    Point mir2 = mir(pol[2], mid), mir1 = mir(pol[1], mid), mir0 = mir(pol[0], mid);
    cout << fixed << setprecision(5) << pol[0].x << '\u' << pol[0].y << '\n';
    cout << fixed << setprecision(5) << mir2.x << '\u' << mir2.y << '\n';
    cout << fixed << setprecision(5) << pol[1].x << '\u' << pol[1].y << '\n';
    cout << fixed << setprecision(5) << mir0.x << '\u' << mir0.y << '\n';
    cout << fixed << setprecision(5) << pol[2].x << '\u' << pol[2].y << '\n';
    cout << fixed << setprecision(5) << mir1.x << '\u' << mir1.y << '\n';
}

```

Task C ()

```
#ifdef _DEBUG
#define _CRT_SECURE_NO_WARNINGS
#endif
#include <iostream>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <algorithm>
#include <queue>
#include <iomanip>
#include <cmath>

using namespace std;

string crr;
int dp[510][10010];

int resolve()
{
    int mx = 0;
    string s;
    cin >> s;
    s = '$' + s;
    for (int i = 1; i < crr.size(); i++)
    {
        for (int l = 1; l < s.size(); l++)
        {
            dp[i][l] = 0;
            if (crr[i] == s[l])
                dp[i][l] = dp[i - 1][l - 1] + 1;
            dp[i][l] = max(dp[i][l], dp[i - 1][l]);
            mx = max(mx, dp[i][l]);
        }
    }
    return crr.size() - mx - 1;
}

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif
    cin >> crr;
    crr = '$' + crr;
    int t;
    cin >> t;
    int ans = 0;
    for (int i = 1; i <= t; i++)
        ans += resolve();
    cout << ans;
}
```

Task D ()

```
#ifdef _DEBUG
#define _CRT_SECURE_NO_WARNINGS
#endif
#include <iostream>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <algorithm>
#include <queue>
#include <iomanip>
#include <cmath>

using namespace std;

int d[51][51];
pair<int, int> dir[51][51];
bool used[51][51];

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    int n, m;
    cin >> n >> m;
    int stR, stC, fnR, fnC;
    cin >> stR >> stC >> fnR >> fnC;
    for (int i = 1; i <= n; i++)
        for (int l = 1; l <= m; l++)
        {
            cin >> dir[i][l].first >> dir[i][l].second;
            d[i][l] = 1e9 + 7;
        }
    d[stR][stC] = 0;
    for (int t = 1; t <= n * m; t++)
    {
        int r = -1, c;
        int minD = 1e9 + 7;
        for (int i = 1; i <= n; i++)
        {
            for (int l = 1; l <= m; l++)
            {
                if (d[i][l] < minD && !used[i][l])
                {
                    minD = d[i][l];
                    r = i;
                    c = l;
                }
            }
        }
        if (r == -1)
            break;
        for (int i = 1; i <= n; i++)
            for (int l = 1; l <= m; l++)
                d[i][l] = min(d[i][l], d[r][c] + abs(i - r - dir[r][c].first) +
                               abs(l - c - dir[r][c].second));
        used[r][c] = 1;
    }
    cout << d[fnR][fnC];
}
```

Task E ()

```
#ifdef _DEBUG
#define _CRT_SECURE_NO_WARNINGS
#endif
#include <iostream>
#include <vector>
#include <string>
#include <set>
#include <map>
#include <algorithm>
#include <queue>
#include <iomanip>
#include <cmath>

using namespace std;

bool used[19000];
long long ONE = 1;

int main()
{
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
#endif

    long long n, m, b;
    cin >> n >> m >> b;
    vector<pair<long long, long long>> v(b);
    for (long long i = 0; i < b; i++)
    {
        cin >> v[i].first >> v[i].second;
        v[i].first--;
        v[i].second--;
    }
    for (long long i = 0; i < b; i++)
    {
        long long quan = 0;
        long long now = 0;
        for (long long num = 0; num < (ONE << (b + 1)); num++)
        {
            if (used[num])
                continue;
            if (now == 0)
                cout << "?_";
            cout << v[i].first + num * n << '__' << v[i].second << '__';
            now++;
            if (now == 2)
            {
                cout << endl;
                long long x, y;
                cin >> x >> y;
                used[x / n] = 1;
                quan += now;
                now = 0;
            }
            if (quan >= (1 << (b - i)))
                break;
        }
        if (now == 1)
            return -1;
    }
    for (long long i = 0; i < (ONE << b); i++)
    {
        if (!used[i])
        {
            cout << "!__" << i * n << '__' << 0 << endl;
            return 0;
        }
    }
    return -1;
}
```

Task F ()