

# Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	30	0	0	330

## Task A ()

```
n = int(input())

def ceil(a, b):
    if a % b != 0:
        return a // b + 1
    else:
        return a // b

print(n-1)
```

## Task B ()

```
n = int(input())

from math import *

def dist(x, y):
    return sqrt((y[0]-x[0])**2 + (y[1] - x[1])**2)

def makevec(fr, to):
    return (to[0] - fr[0], to[1] - fr[1])

if n == 6:
    x, y = list(), list()

    for i in range(6):
        xx, yy = [float(mmm) for mmm in input().split()]
        x.append(xx)
        y.append(yy)

    xc, yc = sum(x)/6, sum(y)/6

    order = list(range(6))

    def Key(i):
        X, Y = makevec((xc, yc), (x[i], y[i]))
        return atan2(Y, X)
    order.sort(key = Key)

    # for i in order:
    #     X, Y = makevec((xc, yc), (x[i], y[i]))
    #     print(atan2(Y, X))

    for i in order[:3]:
        print(x[i], y[i])

elif n == 3:
    x, y = list(), list()

    for i in range(3):
        xx, yy = [float(mmm) for mmm in input().split()]
        x.append(xx)
        y.append(yy)

    cc = ((x[0]+x[2])/2, (y[0]+y[2])/2)
    ddd = makevec((x[1], y[1]), cc)

    c = (cc[0]+ddd[0], cc[1]+ddd[1])
    #print(c, cc)

    for i in range(3):
        print(x[i], y[i])
        vec = makevec(c, (x[i], y[i]))
        x[i], y[i] = c[0] - vec[0], c[1] - vec[1]
    for i in range(3):
        print(x[i], y[i])
    # x[i], y[i] = c[0] - x[i], c[1] - y[i]
```

## Task C ()

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

using namespace std;

int solve(string a, string b) {
    vector<vector<int>> dp(a.length(), vector<int>(b.length(), 0));
    int ans = 0;
    for (int j = 0; j < b.length(); j++) {
        int mx = 0;
        for (int i = 0; i < a.length(); i++) {
            if (i > 0 && j > 0)
                mx = max(mx, dp[i-1][j-1]);
            if (a[i] == b[j]) {
                dp[i][j] = mx + 1;
                ans = max(ans, dp[i][j]);
            }
            //cout << dp[i][j] << " ";
        }
        //cout << endl;
    }
    return ans;
}

int main() {
    int n;
    string t;
    cin >> t;
    cin >> n;
    int ans = 0;
    for (int i = 0; i < n; i++) {
        string tmp;
        cin >> tmp;
        int s = solve(t, tmp);
        //cout << s << endl;
        ans += t.length() - s;
    }
    cout << ans << endl;

    return 0;
}
```

## Task D ()

```
#include<iostream>
#include<vector>
#include<cmath>

#define fs first
#define sc second
#define pb push_back

using namespace std;

int dist(int x1, int y1, int x2, int y2) {
    return abs(x1-x2) + abs(y1-y2);
}

int main() {
    int n, m;
    cin >> n>> m;

    int sx, sy, fx, fy;
    cin >> sx >> sy >> fx >> fy;

    vector<vector<pair<int, int>>> field(n+2, vector<pair<int, int>>(m+2, {1e10, 1e10} ));
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= m; j++)
            cin >> field[i][j].fs >> field[i][j].sc;

    vector<vector<int>> dp (n+2, vector<int>(m+2, -1));

    for (int i = 0; i < n+2; i++) {
        dp[i][0] = -666;
        dp[i][m+1] = -666;
    }
    for (int j = 0; j < m+2; j++) {
        dp[0][j] = -666;
        dp[n+1][j] = -666;
    }

    //sx++; sy++; fx++; fy++;

    vector<vector<pair<int, int>>> ls (n*m, vector<pair<int, int>>());
    ls[0].pb({fx, fy});
    dp[fx][fy] = 0;

    if (fx == sx && fy == sy) {
        cout << 0;
        exit(0);
    }

    int now = 0;
    int done = 0;

    while (now < n*m) {
        int sz = ls[now].size();
        for (int kkk = 0; kkk < sz; kkk++) {
            auto el = ls[now][kkk];
            int i = el.fs, j = el.sc;
            for (int ii = i - 1; ii <= i+1; ii++) {
                for (int jj = j - 1; jj <= j+1; jj++) {
                    if (dp[ii][jj] != -1 or (ii == i && jj == j))
                        continue;
                    if (field[ii][jj].fs + ii == i && field[ii][jj].sc + jj == j) {
                        dp[ii][jj] = dp[i][j];
                        sz++;
                        ls[now].pb({ii, jj});
                        if (ii == sx && jj == sy) {
                            cout << now;
                            exit(0);
                        }
                    }
                }
            }
        }
    }
```

```

    }
}

//cout << "post" << endl;
//for (auto el: ls[now]) {
//    cout << "(" << el.fs << " " << el.sc << ")" ";
//}
//cout << endl;
now ++;
for (int i = now-1; i >= 0; i--) {
    for (pair<int, int> el: ls[i]) {
        for (int I = max(1, el.fs-2-(now-i)); I <= min(n+1, el.fs+2+(now-i)); I++) {
            for (int J = max(1, el.sc-2-(now-i)); J <= min(m+1, el.sc+2+(now-i)); J++) {
                if (dp[I][J] != -1)
                    continue;
                int dx = abs(field[I][J].fs + I - el.fs);
                int dy = abs(field[I][J].sc + J - el.sc);
                if (dx + dy == now-i && (I != el.fs || J != el.sc)) {
                    dp[I][J] = now;
                    ls[now].pb({I, J});
                    if (I == sx && J == sy) {
                        cout << now;
                        exit(0);
                    }
                }
            }
        }
    }
}

//cout << "pre" << endl;
//for (auto el: ls[now]) {
//    cout << "(" << el.fs << " " << el.sc << ")" ";
//}
//cout << endl;

}

return 0;
}

```

## Task E ()

## Task F ()