

**Решения задач**  
**призерами финала номинации «Алгоритмическое программирование»**  
**Поволжской олимпиады «Волга-ИТ – 2015»**

**Тройка призеров:**

- 1. Кондратьев Евгений Валерьевич – 2 задачи, 138 минут**
- 2. Станкевич Артем Сергеевич – 2 задачи, 261 минуты**
- 3. Горшков Даниил Александрович – 2 задачи, 347 минут**

**Задача D: Это будет гроб**

**Лимит времени** 2000/2000/2000/2000 мс.

**Лимит памяти** 65536/65536/65536/65536 Кб.

**ограничение по времени на тест** 2 секунды

**ограничение по памяти на тест** 64 мегабайта

**ввод** стандартный ввод

**вывод** стандартный вывод

Придумать по-настоящему сложную и интересную задачу для финала — дело очень непростое. Вот уже вторую неделю Андрей и Владимир просматривают e-maxx и перечитывают Кнута с Корменом в поисках такого алгоритма, который достоин стать темой самой сложной задачи. Всем известно, что по-настоящему сложные алгоритмы называют в честь нескольких людей, указывая их фамилии через дефис. Андрей и Владимир полагают, что алгоритм должен быть тем сложнее, чем больше людей указано в его названии. Однако фамилии некоторых учёных сами по себе являются двойными и пишутся через дефис, что иногда приводит к путанице (например, у деревьев Адельсона-Вельского-Ландиса двое авторов, а не трое).

Сегодня Владимир показал Андрею по-настоящему сложный алгоритм, в названии которого фигурировало много фамилий. Теперь члены жюри хотят определить минимальное и максимальное количество людей, в честь которых мог быть назван алгоритм. Андрей и Владимир составили список известных им фамилий учёных; если некоторая фамилия из списка состоит из одной части (пишется без дефиса), то она точно не входит в состав никаких двойных фамилий, а если фамилия из списка двойная, то её части точно не являются самостоятельными фамилиями и не входят в состав других двойных фамилий.

Помогите Андрею и Владимиру узнать, сколько авторов было у алгоритма.

**Входные данные**

Первая строка содержит название алгоритма, состоящее из латинских букв и дефисов и имеющее длину от 1 до 1000 символов. Название не начинается и не оканчивается дефисом, а также не содержит двух дефисов подряд. Первая буква каждой части названия является заглавной, остальные буквы — строчными.

Вторая строка содержит целое число  $N$  ( $0 \leq N \leq 100$ ) — количество фамилий учёных, известных Андрею и Владимиру.

Следующие  $N$  строк содержат фамилии учёных, каждая из которых состоит из латинских букв, а также не более одного дефиса, и имеет длину от 1 до 40 символов. Каждая из фамилий не начинается и не оканчивается дефисом. Первая буква каждой фамилии или её части является заглавной, остальные буквы — строчными.

Все указанные во входных данных фамилии и части фамилий различны.

**Выходные данные**

Выведите два целых числа — минимально возможное и максимально возможное количество людей, в честь которых назван алгоритм.

**Примеры тестов**

<b>Тест</b>	<b>Входные данные</b>	<b>Выходные данные</b>
1	Farach-Colton-Bender	2 2

	3 Dijkstra Farach-Colton Ukkonen	
2	Arlazarov-Dinic-Kronrod-Faradzev 0	2 4

### Кондратьев Евгений Валерьевич: [C++]

```

#include <stdio.h>
#include <iostream>
#include <stack>
#include <vector>
#include <set>
#include <map>
#include <queue>
#include <stdlib.h>
#include <algorithm>
#include <string>
#include <string.h>

#define long long int lli;

const int MXN = 200000;

using namespace std;

int n, m, k;
string s[MXN];
string dd[MXN];
string d;
int u[MXN];

void main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);

    cin >> d;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> s[i];
    }

    int sz = d.size();
    string tt = "";
    int q = 0;
    for (int i = 0; i < sz; i++) {
        if (d[i] == '-') {
            dd[q++] = tt;
            tt = "";
            i++;
        }
        tt += d[i];
    }
    dd[q++] = tt;

    string tmp = "";
    int r = 0;
    for (int i = 0; i < q; i++) {
        for (int j = 0; j < n; j++) {
            if (dd[i] == s[j]) {
                u[i] = 1;
                r++;
                break;
            }
        }
        if (r == 0 && i > 0) {

```

```

        if (dd[i - 1] + '-' + dd[i] == s[j]) {
            u[i - 1] = 2;
            u[i] = 2;
        }
    }
}
if (r == 1)
    r = 0;
}

int ans1 = 0;
for (int i = 0; i < q; i++) {
    if (u[i] == 0) {
        ans1++;
    }
    if (u[i] == 1) {
        ans1++;
    }
    if (u[i] == 2) {
        ans1++;
        i++;
    }
}

int ans2 = 0;
int w = 0;
for (int i = 0; i < q; i++) {
    w = 0;
    while (i < q && u[i] == 0) {
        w++;
        i++;
    }
    ans2 += w / 2 + w % 2;
    if (u[i] == 1)
        ans2++;
    if (u[i] == 2) {
        ans2++;
        i++;
    }
}

cout << ans2 << " " << ans1;
}

```

### **Станкевич Артём Сергеевич[C++]**

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <iomanip>
#include <algorithm>
#include <vector>
#include <queue>
#include <stack>
#include <string>
#include <set>
#include <map>

#define ll long long int

using namespace std;
const int INF = 1000000000, N = 200;
vector<string > t(N);
string s;
int n, def = 0, mn = 0, mx = 0;
vector<pair<int, bool > > df, f;

```

```

int main(){
    //freopen("input.txt","r",stdin); freopen("output.txt","w",stdout);
    cin >> s >> n;
    for(int i =0;i<n;++i)
        cin >> t[i];

    for(int i = 0; i<s.length(); ++i)
        if(s[i] == '-') def++;

    for(int i = 0;i<n;++i){
        bool flag = false;
        int u = 0, k = 0;
        int pp = 0, ppp = 0;
        for(int j = 0;j<(int)s.length();++j){
            if(u<(int)t[i].length() && s[j] == t[i][u]) {if(t[i][u++] == '-')
') flag = true;}
            else u = 0;
            if(s[j] =='-') {k++; ppp = pp; pp = j;}
            if(u == t[i].length() && (j==s.length()-1 || j+1<s.length()&&s[j+1]=='-
') && (
                (ppp==0)&&(j+1==t[i].length()&&!flag || flag && j+1==t[i].length() ) ||
                (j-pp==t[i].length()&&!flag || flag && j-ppp==t[i].length() ) )
                df.push_back(make_pair(max(k-flag,0), flag));

        }
    }

    sort(df.begin(),df.end());
    int p = 0;
    for(int i =0;i<(int)df.size();++i){
        int to = df[i].first-p;
        mx+=to+1;
        mn+=to/2+to%2+1;
        p = df[i].first+1;
        if(df[i].second) p++;
    }

    if(df.size()==0) {
        mn = (def+1)/2+(def+1)%2;
        mx = def+1;
    }
    else{
        mx+=(def+1-p);
        mn+=(def+1-p)/2+(def+1-p)%2;
    }
    cout << mn << " " << mx<< endl;
    return 0;
}

```

### **Горшков Даниил Александрович[C++]**

```

#include <iostream>
#include <cmath>
#include <string>
#include <string.h>
#include <cstring>
#include <algorithm>
#include <stdio.h>
using namespace std;

typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;
//ll x, y, z;
//ll A[100000], B[100000];
//int a, b;

```

```

struct slovo{
    char p[50];
    char v[50];
};
int pm(char h[]){
    int f=strlen(h);
    for (int i=0; i < f;i++){
        if (h[i]=='-') return i;
    }
    return 0;
}
int a = 0, b=0;
char S[110][50], A[110][50],temp[50];
char C[1100];
slovo B[110];

bool provdv(char h[]){
    for(int i = 0; i<b; i++){
        if(!strcmp(h, B[i].p)) return true;
    }
    return false;
}
bool provod(char h[]){
    for(int i = 0; i<a; i++){
        if(!strcmp(h, A[i])) return true;
    }
    return false;
}
int main()
{
    cin >> C;

    int k= strlen(C);
    for (int i=0; i < k;i++){
        if (C[i] != '-'){
            S[a][b] = C[i];
            b++;
        } else {
            S[a][b] ='\0';
            a++;
            b=0;
        }
    }
    S[a][b]='\0';
    int razmer = a+1;
    int n;
    cin >>n;
    a=0; b=0;
    for (int i=0; i < n;i++){
        cin>>temp;
        int kk = strlen(temp);
        k = pm(temp);
        if(k){
            for (int i = 0; i < k;i++){
                B[b].p[i] = temp[i];
            }
            for (int i = k+1; i < kk;i++){
                B[b].v[i-k-1] = temp[i];
            }
            b++;
        } else{
            for (int i = 0; i < kk;i++){
                A[a][i] = temp[i];
            }
            a++;
        }
    }
}

```

```

    }
}
int min=0, max=0;
for(int i = 0; i < razmer; i++){
    if(provdv(S[i])){
        max++;
        i++;
    } else{
        max++;
    }
}for(int i = 0; i < razmer; i++){
    if(provdv(S[i])){
        min++;
        i++;
        continue;
    }
    if (provod(S[i])){
        min++;
        continue;
    }
    if (i == (razmer - 1)) {
        min++;
        continue;
    }
    if((provdv(S[i+1]))||(provod(S[i+1]))){
        min++;
        continue;
    }
    min++;
    i++;
}
if ((min < (razmer/2))||(max>razmer)) {cout << "kkk";return -1;}
cout << min <<" "<<max;
return 0;
}

```

## Задача F: F.A.T.S

Лимит времени 2000/2000/2000/2000 мс.

Лимит памяти 65536/65536/65536/65536 Кб.

ограничение по времени на тест 2 секунды

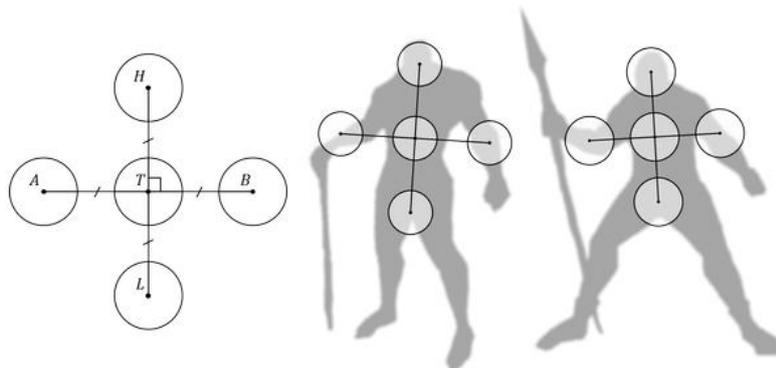
ограничение по памяти на тест 64 мегабайта

ввод стандартный ввод

вывод стандартный вывод

В RawCraft испытывают систему пошагового прицеливания F.A.T.S.

(FreakyAssistedTargetingSystem), идею которой разработчики подглядели в какой-то другой игре.



Для каждого персонажа система отслеживает положение точек  $H$ ,  $T$ ,  $L$ ,  $A$ ,  $B$  — соответственно головы, туловища, ног, правой и левой руки. Длины отрезков  $HT$ ,  $LT$ ,  $AT$ ,  $BT$  одинаковы, отрезки  $HL$  и  $AB$  перпендикулярны. Если снаряд или заклинание попадают в круг радиуса  $R$ , центром которого является одна из отслеживаемых точек, то персонаж получает повреждение соответствующей части тела:

- Попадание в голову приносит 3 очка урона;
- Попадание в туловище приносит 2 очка урона;
- Попадание в остальные части тела приносит 1 очко урона.

Вася пользуется системой F.A.T.S., чтобы точнее стрелять огненными заклинаниями. Встретив противника, Вася сколдовал  $N$  фаерболлов. Определите, сколько урона получил от них противник.

### Входные данные

Первая строка содержит целые числа  $X_H, Y_H, X_T, Y_T$  и  $R$  ( $-1000 \leq X_H, Y_H, X_T, Y_T \leq 1000$ ,  $1 \leq R < \frac{HT}{2}$ ) — соответственно координаты точки  $H$ , координаты точки  $T$  и радиус области повреждения частей тела.

Вторая строка содержит целое число  $N$  ( $1 \leq N \leq 1000$ ) — количество заклинаний Васи.

Каждая из следующих  $N$  строк содержит целые числа  $X_i, Y_i$  ( $-1000 \leq X_i, Y_i \leq 1000$ ) — координаты точки, в которую попало  $i$ -е заклинание.

### Выходные данные

Выведите одно целое число — общее количество очков урона, полученных противником Васи.

### Примеры тестов

Тест	Входные данные	Выходные данные
1	0 3 0 0 1 3 0 2 2 2 -3 -1	4
2	-2 4 0 0 2 2 3 -3 2 4	1

```

#include <stdio.h>
#include <iostream>
#include <stack>
#include <vector>
#include <set>
#include <map>
#include <queue>
#include <stdlib.h>
#include <algorithm>
#include <string>
#include <string.h>

#define long long int lli;

const int MXN = 200000;

using namespace std;

int n, m, k, x, y, x0, qy0, r, xa, xb, xc, ya, yb, yc;

int vx[MXN];
int vy[MXN];

void main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);

    cin >> x >> y >> x0 >> qy0 >> r;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> vx[i] >> vy[i];
    }

    //dx = max(dx, -dx);
    //dy = max(dy, -dy);
    int dx = x - x0;
    int dy = y - qy0;

    xa = x0 - dy;
    ya = qy0 + dx;

    xb = x0 - dx;
    yb = qy0 - dy;

    xc = x0 + dy;
    yc = qy0 - dx;

    int ans = 0;
    for (int i = 0; i < n; i++) {
        if ((vx[i] - xa)*(vx[i] - xa) + (vy[i] - ya)*(vy[i] - ya) <= r*r) {
            ans++;
        }
        if ((vx[i] - xb)*(vx[i] - xb) + (vy[i] - yb)*(vy[i] - yb) <= r*r) {
            ans++;
        }
        if ((vx[i] - xc)*(vx[i] - xc) + (vy[i] - yc)*(vy[i] - yc) <= r*r) {
            ans++;
        }
        if ((vx[i] - x)*(vx[i] - x) + (vy[i] - y)*(vy[i] - y) <= r*r) {
            ans+=3;
        }
        if ((vx[i] - x0)*(vx[i] - x0) + (vy[i] - qy0)*(vy[i] - qy0) <= r*r) {
            ans+=2;
        }
    }
}

```

```

    }
}

cout << ans;

}

```

### Станкевич Артём Сергеевич[C++]

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <iomanip>
#include <algorithm>
#include <vector>
#include <queue>
#include <stack>
#include <string>
#include <set>
#include <map>

#define ll long long int

using namespace std;
const int INF = 1000000000, N = 200;
int hx, hy, tx, ty, lx, ly, ax, ay, bx, by, r, bon = 0, n;

bool pr(int x, int y, int xx, int yy){
    return (x-xx)*(x-xx)+(y-yy)*(y-yy) <= r*r;
}

int main(){
    //freopen("input.txt", "r", stdin); freopen("output.txt", "w", stdout);
    cin >> hx >> hy >> tx >> ty >> r >> n;

    int vx = (hx-tx), vy = (hy-ty);
    lx = tx-vx;
    ly = ty-vy;
    ax = tx-vy;
    ay = ty+vx;
    bx = tx+vy;
    by = ty-vx;
    for(int i = 0; i < n; ++i){
        int x, y;
        cin >> x >> y;
        if(pr(lx, ly, x, y) || pr(ax, ay, x, y) || pr(bx, by, x, y)) bon++;
        if(pr(tx, ty, x, y)) bon+=2;
        if(pr(hx, hy, x, y)) bon+=3;
    }
    cout << bon << endl;
    return 0;
}

```

### Горшков Даниил Александрович[C++]

```

#include <iostream>
#include <cmath>
#include <string>
#include <cstring>
#include <algorithm>
using namespace std;

typedef long long ll;
typedef unsigned long long ull;

//ll x, y, z;

```

```

//ll A[100000], B[100000];
//int a, b;

struct koord{
    int x;
    int y;
};

bool prov(koord x, koord y, int r){
    int rr = r*r;

    // cout << x.x <<" "<<x.y <<" "<<y.x <<" "<<y.y <<endl;
    long double d = pow(abs(x.x - y.x),2.0)+pow(abs(x.y - y.y),2.0);
    // cout << d;
    if(d<=rr){
        return true;
    }
    return false;
}

int main()
{
    int r, n, k =0;
    koord h, t, l, a, b, m[10000], v, va, vb;
    cin >> h.x >> h.y>>t.x>>t.y>>r;
    cin >> n;
    for (int i = 0; i<n;i++){
        cin >>m[i].x>>m[i].y;
    }
    v.x=t.x - h.x;
    v.y=t.y - h.y;
    va.x= -v.y;
    va.y= v.x;
    vb.x= v.y;
    vb.y= -v.x;

    l.x= t.x + v.x;
    l.y= t.y + v.y;
    a.x= t.x + va.x;
    a.y= t.y + va.y;
    b.x= t.x + vb.x;
    b.y= t.y + vb.y;

    for (int i = 0; i<n;i++){
        if (prov(m[i], h,r)) k+=3;
        if (prov(m[i], t,r)) k+=2;
        if (prov(m[i], l,r)) k+=1;
        if (prov(m[i], a,r)) k+=1;
        if (prov(m[i], b,r)) k+=1;
    }
    cout << k;
    return 0;
}

```