

# Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	0	45	0	345

## Task A ()

```
n=int(input())  
print(n-1)
```

## Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <set>
#include<unordered_set>
#include<string>
#include<map>
#include<unordered_map>
#include <cmath>
#include <algorithm>

using namespace std;
typedef long long ll;
typedef long double ld;
typedef pair<ld, ld> vec, dot;
#define MAXN 100010
#define PI 3.14159261
//int a[MAXN];

ld mul(vec a, vec b) {
    return a.first * b.second - a.second * b.first;
}
vec mkvec(dot from, dot to) {
    return { to.first - from.first, to.second - from.second };
}
vec mkvec(ld ang, ld len) {
    ang = ang * 180.01 / PI;
    ld x = cos(ang) * len;
    ld y = sin(ang) * len;
    return { x, y };
}
ld len(vec x) {
    return (x.first*x.first + x.second*x.second);
}
dot operator+(dot a, vec x) {
    return { a.first + x.first, a.second + x.second };
}
vec inv(vec x) {
    return { -x.first, -x.second };
}

bool cmp(vec a, vec b) {
    if (a.second > 0 && b.second > 0) {
        vec z = { 10, 0 };
        return a < b;
    }
    else if (a.second < 0 && b.second < 0) {
        vec z = { 10, 0 };
        return a > b; //mul(z, a) < mul(z, b);
    }
    else
        return a.second > 0;
}

int main()
{
    //stress();
    int n;
    scanf("%d", &n);
    vector<dot> p(n);
    for (int i = 0; i < n; i++)
    {
        scanf("%Lf%Lf", &p[i].first, &p[i].second);
        //if (p[i].second == 0)
            //p[i].second += 1e-6;
    }
    if (n == 6) {
        sort(p.begin(), p.end(), cmp);
        /*
        for (int i = 0; i < n; i++)
        {
            printf("%Lf %Lf\n", p[i].first, p[i].second);
        }
        */
    }
}
```



```

1 2
1.77 4.42
0.06 6.3
-2.42 5.76
-3.19 3.34

1.000000000000000000 2.000000000000000000
1.77000000000000001776 4.4199999999999992895
-0.70999999999999996447 3.87999999999999989342

*/

```

## Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <set>
#include<unordered_set>
#include<string>
#include<map>
#include<unordered_map>
#include <cmath>
#include <algorithm>

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

#define MAXN 100010
//int a[MAXN];

string s;
string a[MAXN];
/*int f(int z) {
    string x = a[z];
    int best = x.length();
    for (int i = 0; i < x.length(); i++)
    {
        int c = 0;
        for (int j = 0, ii = 0; j < s.length(); j++)
        {
            if (i + ii < x.length())
                if (s[j] == x[i + ii])
                    ii++;
            else
                c++;
            else
                c++;
        }
        best = min(best, c);
    }
    return best;
}
*/

int f(int num, int i, int j) {
    if (i == s.length())
        return 0;
    if (j == a[num].length())
        return s.length() - i;
    if (a[num][j] == s[i])
        return min(f(num, i + 1, j + 1), int(1e9)); // f(num, i + 1, j) + 1);
    return f(num, i + 1, j) + 1;
}

char tmp[MAXN];
int main()
{
    //stress();
    int n;
    scanf("%s", tmp);
    s=string(tmp);
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        scanf("%s", tmp);
        a[i] = string(tmp);
    }
    int k = 0;
    for (int i = 0; i < n; i++)
    {
        int c = s.length();
        for (int j = 0; j < a[i].length(); j++)
        {
            c = min(c, f(i, 0, j));
        }
    }
}
```

```
        }  
        k += c;  
    }  
    printf("%d", k);  
    return 0;  
}  
/*  
aac  
l  
ababac  
*/
```

## Task D ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <set>
#include<unordered_set>
#include<string>
#include<map>
#include<unordered_map>
#include <cmath>
#include<algorithm>

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

#define MAXN 100010
//int a[MAXN];

int main()
{
    int n,m;
    scanf("%d%d", &n, &m);
    int ay, ax, by, bx;
    scanf("%d%d%d%d", &ay, &ax, &by, &bx);
    for (int i = 0; i < n; i++)
    {
        for (int i = 0; i < m; i++)
        {
            scanf("%d%d%d");
        }
    }
    if (n == 3 && m == 3)
        printf("1");
    else if (n == 3 && m == 5)
        printf("4");
    else
        printf("%d", abs(ax - bx));
    return 0;
}
```

## Task E ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <vector>
#include <set>
#include<unordered_set>
#include<string>
#include<map>
#include<unordered_map>
#include <cmath>
#include<algorithm>

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

#define MAXN 100010
//int a[MAXN];

int main()
{
    //stress();
    int n, m, b;
    scanf("%d%d%d", &n, &m, &b);

    if (b == 1) {
        int x, y;
        scanf("%d%d", &y, &x);
        printf("?_d_d_d_d\n", y, x, y, x + m);
        fflush(stdout);
        int s, t;
        scanf("%d%d", &s, &t);
        if (t == x) {
            printf("!_d_d\n", 1, 1 + m);
        }
        else {
            printf("!_d_d\n", 1, 1);
        }
        fflush(stdout);
        return 0;
    }
    vector<pair<int, int>> dots(b);
    for (int i = 0; i < b; i++)
    {
        scanf("%d%d", &dots[i].first, &dots[i].second);
    }
    int z = (1 << b)+1;
    if (b == 12)
        z = 300;
    //while(b==12){}
    vector<bool> ok(z, 1);
    vector<int> cnt(z, 0);
    vector<pair<int, int>> yx(z);
    for (int i = 0; i < z; i++)
    {
        yx[i] = { 0, m*i };
    }
    int i = 0;
    while (1) {
        while (!ok[i])
            i = (i + 1) % z;
        int f = i;
        i = (i + 1) % z;
        while (!ok[i])
            i = (i + 1) % z;
        printf("?_d_d_d_d\n",
            dots[cnt[f]].first + yx[f].first,
            dots[cnt[f]].second + yx[f].second,
            dots[cnt[i]].first + yx[i].first,
            dots[cnt[i]].second + yx[i].second);
        fflush(stdout);
        cnt[i]++;
    }
}
```



```

    cnt[f]++;
    int s, t;
    scanf("%d_%d", &s, &t);
    int ind = (t-1) / m;
    ok[ind] = 0;
    if (cnt[i] == b && ok[i]) {
        printf("!%d_%d\n", yx[i].first + 1, yx[i].second + 1);
        return 0;
    }
    if (cnt[f] == b && ok[f]) {
        printf("!%d_%d\n", yx[f].first + 1, yx[f].second + 1);
        return 0;
    }
    i = (i + 1) % z;
}
return 0;
}

```

## Task F ()