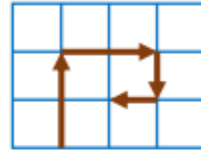


## Bài 12: Kéo cắt giấy

Để trang trí phòng phục vụ tổ chức sinh nhật cho một người bạn Alice lấy một tờ giấy màu thủ công kẻ ô vuông kích thước  $n \times m$  ( $n$  hàng và  $m$  cột), cắt thành hình lò xo xoắn theo hướng phải sang trái và có độ rộng của đường bằng 1:

- Bắt đầu từ biên phải cột 0 cất lên trên cho đến khi cách lề trên một ô,
- Cất sang phải theo đường biên dưới cho đến khi cách lề phải một ô,
- Cất xuống dưới, rồi sang trái, sau đó lên trên, . . . để có băng giấy độ rộng 1 ô,
- Quá trình cất sẽ dừng khi không cách cất tiếp mà không làm đứt băng giấy.



Hãy tính tổng độ dài đường cắt theo đơn vị ô.

**Dữ liệu:** Vào từ file SCISSOR.INP gồm một dòng chứa 2 số nguyên  $n$  và  $m$  ( $2 \leq n, m \leq 10^9$ ).

**Kết quả:** Đưa ra file văn bản SCISSOR.OUT một số nguyên – độ dài đường cắt.

***Ví dụ:***

SCISSOR.INP	SCISSOR.OUT
3 4	6

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "SCISSOR";

#define int long long

int n,m;

int32_t main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    freopen( (filename + ".inp").c_str(), "r", stdin);
    freopen( (filename + ".out").c_str(), "w", stdout);

    cin >> n >> m;
    int x,y;
    int cnt_x = m-1;
    int tmp = n-1;
    x = tmp * (tmp + 1) / 2;
    if (tmp > cnt_x)
    {
        int z = (tmp - cnt_x);
        x -= z * (z + 1) / 2;
    }
    int cnt_y = n-1;
    tmp = m-2;
    y = tmp * (tmp + 1) / 2;
    if (tmp > cnt_y)
```

```
{
    int z = (tmp - cnt_y);
    y -= z * (z + 1) / 2;
}
cout << x + y;
```

```
}
```

## Bài 13: Tập lớn nhất

Cho dãy số nguyên dương  $\mathbf{A} = (a_1, a_2, \dots, a_n)$ . Hãy tìm tập nhiều phần tử nhất có cùng ít nhất một ước số chung lớn hơn 1 và đưa ra số phần tử trong tập tìm được.

Ví dụ, với  $\mathbf{A} = (6, 15, 10, 42)$ , tập  $\{6, 10, 42\}$  chứa các số cùng chia hết cho 2 và là tập nhiều phần tử nhất có cùng ít nhất một ước số chung lớn hơn 1. Số lượng các phần tử trong tập là 3.

**Dữ liệu:** Vào từ file MAX\_SET.INP:

- Đòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 1000$ ),
- Đòng thứ 2 chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^{18}$ ,  $i = 1 \div n$ ).

**Kết quả:** Đưa ra file văn bản MAX\_SET.OUT một số nguyên – số lượng phần tử của tập tìm được.

**Ví dụ:**

MAX_SET.INP
4
6 15 10 42

MAX_SET.OUT
3

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "MAX_SET";

typedef long long ll;
#define MAXL (50000>>5)+1
#define GET(x) (mark[x>>5]>>(x&31)&1)
#define SET(x) (mark[x>>5] |= 1<<(x&31))
int mark[MAXL];
int P[50000], Pt = 0;
void sieve() {
    register int i, j, k;
    SET(1);
    int n = 46340;
    for (i = 2; i <= n; i++) {
        if (!GET(i)) {
            for (k = n/i, j = i*k; k >= i; k--, j -= i)
                SET(j);
            P[Pt++] = i;
        }
    }
}

long long mul(unsigned long long a, unsigned long long b, unsigned long long mod)
{
    long long ret = 0;
    for (a %= mod, b %= mod; b != 0; b >>= 1, a <<= 1, a = a >= mod ? a - mod :
a) {
        if (b&1) {
            ret += a;
            if (ret >= mod) ret -= mod;
        }
    }
    return ret;
}

void exgcd(long long x, long long y, long long &g, long long &a, long long &b) {
```

```

    if (y == 0)
        g = x, a = 1, b = 0;
    else
        exgcd(y, x%y, g, b, a), b -= (x/y) * a;
}
long long llgcd(long long x, long long y) {
    if (x < 0) x = -x;
    if (y < 0) y = -y;
    if (!x || !y) return x + y;
    long long t;
    while (x%y)
        t = x, x = y, y = t%y;
    return y;
}
long long inverse(long long x, long long p) {
    long long g, b, r;
    exgcd(x, p, g, r, b);
    if (g < 0) r = -r;
    return (r%p + p)%p;
}
long long mpow(long long x, long long y, long long mod) { // mod < 2^32
    long long ret = 1;
    while (y) {
        if (y&1)
            ret = (ret * x)%mod;
        y >>= 1, x = (x * x)%mod;
    }
    return ret % mod;
}
long long mpow2(long long x, long long y, long long mod) {
    long long ret = 1;
    while (y) {
        if (y&1)
            ret = mul(ret, x, mod);
        y >>= 1, x = mul(x, x, mod);
    }
    return ret % mod;
}
int isPrime(long long p) { // implements by miller-babin
    if (p < 2 || !(p&1)) return 0;
    if (p == 2) return 1;
    long long q = p-1, a, t;
    int k = 0, b = 0;
    while (!(q&1)) q >>= 1, k++;
    for (int it = 0; it < 2; it++) {
        a = rand()%(p-4) + 2;
        t = mpow2(a, q, p);
        b = (t == 1) || (t == p-1);
        for (int i = 1; i < k && !b; i++) {
            t = mul(t, t, p);
            if (t == p-1)
                b = 1;
        }
        if (b == 0)
            return 0;
    }
    return 1;
}
long long pollard_rho(long long n, long long c) {

```

```

    long long x = 2, y = 2, i = 1, k = 2, d;
    while (true) {
        x = (mul(x, x, n) + c);
        if (x >= n) x -= n;
        d = llgcd(x - y, n);
        if (d > 1) return d;
        if (++i == k) y = x, k <= 1;
    }
    return n;
}

void factorize(int n, vector<long long> &f) {
    for (int i = 0; i < Pt && P[i]*P[i] <= n; i++) {
        if (n%P[i] == 0) {
            while (n%P[i] == 0)
                f.push_back(P[i]), n /= P[i];
        }
    }
    if (n != 1) f.push_back(n);
}

int T;

void llfactorize(long long n, vector<long long> &f) {
    if (T == 50)
        return ;
    T++;
    if (n == 1)
        return ;
    if (n < 1e+9) {
        factorize(n, f);
        return ;
    }
    if (isPrime(n)) {
        f.push_back(n);
        return ;
    }
    long long d = n;
    for (int i = 2; d == n; i++)
        d = pollard_rho(n, i);
    llfactorize(d, f);
    llfactorize(n/d, f);
}

map<ll,int> cnt;

int32_t main()
{
    srand(time(NULL));

    freopen( (filename + ".inp").c_str(), "r", stdin);
    freopen( (filename + ".out").c_str(), "w", stdout);

    sieve();
    int n;
    cin >> n;
    for (int i=1;i<=n;i++)
    {
        T = 0;

```

```

        ll x;
        cin >> x;
        vector<ll> tmp;
        llfactorize(x,tmp);
        sort(tmp.begin(),tmp.end());
        vector<ll>::iterator it = unique(tmp.begin(),tmp.end());
        tmp.resize(distance(tmp.begin(),it));
        for (int j=0;j<tmp.size();j++)
        {
            //cerr << tmp[j] << ' ';
            cnt[tmp[j]]++;
        }
        //cerr << '\n';
    }
    int res = 0;
    for (auto&p : cnt)
        res = max(res,p.second);
    cout << res;
    return 0;
}

```



## Bài 14: Mắt xích yêu nhất

Đoàn thanh niên tổ chức một cuộc chạy kêu gọi mọi người hưởng ứng phong trào “*Vì một cuộc sống không có rác thải nhựa*”. Có  $n$  người tham gia cuộc chạy, mỗi người mặc một áo phông có in số ở lưng, áo người thứ  $i$  có số là  $a_i$ ,  $i = 1 \div n$ .

Sau cuộc chạy mọi người tập trung ở quảng trường thành phố, tham gia trò chơi tập thể “*Khâu yếu nhất*”. Các bạn đứng thành một vòng tròn, cạnh người thứ 2 là người thứ nhất và thứ 3, cạnh người thứ 3 là người thứ 2 và thứ tư, . . ., cạnh người thứ  $n$  là người thứ  $n-1$  và người thứ nhất. Trò chơi bao gồm nhiều lượt đi. Ở mỗi lượt, những ai có số áo nhỏ hơn số áo hai người cạnh mình bước ra khỏi hàng, những người còn lại đứng dồn khít thành vòng tròn nhỏ hơn. Trò chơi kết thúc khi trong vòng tròn chỉ còn có 2 người hoặc khi không có ai phải bước ra ngoài.

Với mỗi người hãy xác định lượt đi mà họ phải bước ra ngoài. Những người còn lại trong vòng tròn khi trò chơi kết thúc có số của lượt đi ra là 0. Các lượt đi đánh số từ 1.

**Dữ liệu:** Vào từ file NEXUS.INP:

-  Dòng đầu tiên chứa số nguyên  $n$  ( $2 \leq n \leq 2 \times 10^5$ ),
-  Dòng thứ 2 chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ,  $i = 1 \div n$ ).

**Kết quả:** Đưa ra file văn bản NEXUS.OUT trên một dòng  $n$  số nguyên, số thứ  $i$  xác định lượt đi ra của người thứ  $i$ ,  $i = 1 \div n$ .

**Ví dụ:**

NEXUS.INP					
5					
5	1	3	1	5	

NEXUS.OUT					
0	1	2	1	0	

## Bài 15: Tập lớn nhất

Cho dãy số nguyên dương  $\mathbf{A} = (a_1, a_2, \dots, a_n)$ . Hãy tìm tập nhiều phần tử nhất có cùng ít nhất một ước số chung lớn hơn 1 và đưa ra số phần tử trong tập tìm được.

Ví dụ, với  $\mathbf{A} = (6, 15, 10, 42)$ , tập  $\{6, 10, 42\}$  chứa các số cùng chia hết cho 2 và là tập nhiều phần tử nhất có cùng ít nhất một ước số chung lớn hơn 1. Số lượng các phần tử trong tập là 3.

**Dữ liệu:** Vào từ file MAX\_SET.INP:

- 🚩 Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 1000$ ),
- 🚩 Dòng thứ 2 chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^{18}$ ,  $i = 1 \div n$ ).

**Kết quả:** Đưa ra file văn bản MAX\_SET.OUT một số nguyên – số lượng phần tử của tập tìm được.

**Ví dụ:**

MAX_SET.INP
4
6 15 10 42

MAX_SET.OUT
3

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "MAX_SET";

typedef long long ll;
#define MAXL (50000>>5)+1
#define GET(x) (mark[x>>5]>>(x&31)&1)
#define SET(x) (mark[x>>5] |= 1<<(x&31))
int mark[MAXL];
int P[50000], Pt = 0;
void sieve() {
    register int i, j, k;
    SET(1);
    int n = 46340;
    for (i = 2; i <= n; i++) {
        if (!GET(i)) {
            for (k = n/i, j = i*k; k >= i; k--, j -= i)
                SET(j);
            P[Pt++] = i;
        }
    }
}

long long mul(unsigned long long a, unsigned long long b, unsigned long long mod)
{
    long long ret = 0;
    for (a %= mod, b %= mod; b != 0; b >>= 1, a <= 1, a = a >= mod ? a - mod :
a) {
        if (b&1) {
            ret += a;
            if (ret >= mod) ret -= mod;
        }
    }
    return ret;
}
```



```

void exgcd(long long x, long long y, long long &g, long long &a, long long &b) {
    if (y == 0)
        g = x, a = 1, b = 0;
    else
        exgcd(y, x%y, g, b, a), b -= (x/y) * a;
}

long long llgcd(long long x, long long y) {
    if (x < 0) x = -x;
    if (y < 0) y = -y;
    if (!x || !y) return x + y;
    long long t;
    while (x%y)
        t = x, x = y, y = t%y;
    return y;
}

long long inverse(long long x, long long p) {
    long long g, b, r;
    exgcd(x, p, g, r, b);
    if (g < 0) r = -r;
    return (r%p + p)%p;
}

long long mpow(long long x, long long y, long long mod) { // mod < 2^32
    long long ret = 1;
    while (y) {
        if (y&1)
            ret = (ret * x)%mod;
        y >>= 1, x = (x * x)%mod;
    }
    return ret % mod;
}

long long mpow2(long long x, long long y, long long mod) {
    long long ret = 1;
    while (y) {
        if (y&1)
            ret = mul(ret, x, mod);
        y >>= 1, x = mul(x, x, mod);
    }
    return ret % mod;
}

int isPrime(long long p) { // implements by miller-babin
    if (p < 2 || !(p&1)) return 0;
    if (p == 2) return 1;
    long long q = p-1, a, t;
    int k = 0, b = 0;
    while (!(q&1)) q >>= 1, k++;
    for (int it = 0; it < 2; it++) {
        a = rand()%(p-4) + 2;
        t = mpow2(a, q, p);
        b = (t == 1) || (t == p-1);
        for (int i = 1; i < k && !b; i++) {
            t = mul(t, t, p);
            if (t == p-1)
                b = 1;
        }
        if (b == 0)
            return 0;
    }
    return 1;
}

```

```

long long pollard_rho(long long n, long long c) {
    long long x = 2, y = 2, i = 1, k = 2, d;
    while (true) {
        x = (mul(x, x, n) + c);
        if (x >= n) x -= n;
        d = llgcd(x - y, n);
        if (d > 1) return d;
        if (++i == k) y = x, k <= 1;
    }
    return n;
}

void factorize(int n, vector<long long> &f) {
    for (int i = 0; i < Pt && P[i]*P[i] <= n; i++) {
        if (n%P[i] == 0) {
            while (n%P[i] == 0)
                f.push_back(P[i]), n /= P[i];
        }
    }
    if (n != 1) f.push_back(n);
}

int T;

void llfactorize(long long n, vector<long long> &f) {
    if (T == 50)
        return ;
    T++;
    if (n == 1)
        return ;
    if (n < 1e+9) {
        factorize(n, f);
        return ;
    }
    if (isPrime(n)) {
        f.push_back(n);
        return ;
    }
    long long d = n;
    for (int i = 2; d == n; i++)
        d = pollard_rho(n, i);
    llfactorize(d, f);
    llfactorize(n/d, f);
}

map<ll,int> cnt;

int32_t main()
{
    srand(time(NULL));

    freopen( (filename + ".inp").c_str(), "r", stdin);
    freopen( (filename + ".out").c_str(), "w", stdout);

    sieve();
    int n;
    cin >> n;
    for (int i=1;i<=n;i++)
    {

```

```

        T = 0;
        ll x;
        cin >> x;
        vector<ll> tmp;
        llfactorize(x,tmp);
        sort(tmp.begin(),tmp.end());
        vector<ll>::iterator it = unique(tmp.begin(),tmp.end());
        tmp.resize(distance(tmp.begin(),it));
        for (int j=0;j<tmp.size();j++)
        {
            //cerr << tmp[j] << ' ';
            cnt[tmp[j]]++;
        }
        //cerr << '\n';
    }
    int res = 0;
    for (auto&p : cnt)
        res = max(res,p.second);
    cout << res;
    return 0;
}

```

## Bài 16: Giải thoát

Một thiết bị thăm dò điều khiển từ xa được thả xuống khảo sát mặt đáy của một bể hóa chất hình chữ nhật kích thước  $n \times m$  ô vuông. Thiết bị có thể được điều khiển để đi từ một ô sang ô kề cạnh. Trục chuyển động theo chiều ngang (sang ô cùng hàng) được kết nối với thiết bị làm lạnh, cứ mỗi lần di chuyển sang ô kề cạnh cùng hàng nhiệt độ bên trong thiết bị giảm đi 1. Trục chuyển động theo chiều dọc (sang ô cùng cột) được gắn với thiết bị làm nóng, cứ mỗi lần di chuyển sang ô kề cạnh cùng cột nhiệt độ bên trong thiết bị tăng thêm 1.

Sau khi hoàn thành nhiệm vụ khảo sát thiết bị đang ở ô được đánh dấu là '**s**'. Thật không may ông hút đưa thiết bị lên trên bị kẹt và chỉ có thể thu hồi thiết bị khảo sát nếu nó ở ô được đánh dấu '**f**'. Ngoài ra, khi đưa lên thiết bị cần có nhiệt độ gần 0 nhất có thể.

Tình trạng đáy của bể hóa chất được xác định bởi bản đồ **B** kích thước  $n \times m$ .  $B_{i,j}$  được đánh dấu '**.**' nếu là ô trống và thiết bị thăm dò có thể đi qua. Nếu ô  $(i, j)$  có vật cản, không thể đi vào thì  $B_{i,j}$  được đánh dấu là '**#**'. Tồn tại một ô được đánh dấu '**s**' và một ô khác – đánh dấu '**f**'.

Hãy xác định chênh lệch nhiệt độ tối thiểu (so với 0) thiết bị thăm dò có thể đạt được khi thoát ra khỏi bể.

**Dữ liệu:** Vào từ file ESCAPE.INP:

- 🚩 Dòng đầu tiên chứa 2 số nguyên  $n$  và  $m$  ( $1 \leq n, m \leq 1000$ ),
- 🚩 Dòng thứ  $i$  trong  $n$  dòng sau chứa xâu độ dài  $m$  chứa các ký tự đã nêu, mô tả dòng thứ  $i$  của bản đồ **B**.

**Kết quả:** Đưa ra file văn bản ESCAPE.OUT một số nguyên – chênh lệch nhiệt độ tối thiểu có thể đạt được. Nếu không thể cứu được thiết bị thì đưa ra số -1.

**Kết quả:** Đưa ra file văn bản ESCAPE.OUT một số nguyên – chênh lệch nhiệt độ tối thiểu có thể đạt được. Nếu không thể cứu được thiết bị thì đưa ra số -1.

**Ví dụ:**

ESCAPE.INP
4 3
. . f
. . #
s # #
. . .

ESCAPE.OUT
0

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "ESCAPE";

struct Node
{
    int x,y,dir,turn;
    Node() {}
    Node(int _x,int _y,int _dir,int _turn)
    {
        x = _x;
        y = _y;
        dir = _dir;
```

```

        turn = _turn;
    }
};

int n,m;
char a[1001][1001];
bool rip[1001][1001];
bool visited[1001][1001][4][2];
int s_x,s_y;
int f_x,f_y;
int dx[4] = {-1,0,1,0};
int dy[4] = {0,1,0,-1};

void bfs()
{
    queue<Node> CurrList;
    for (int i=0;i<4;i++)
    {
        int xx = s_x + dx[i];
        int yy = s_y + dy[i];
        if (1 <= xx && xx <= n && 1 <= yy && yy <= m && !rip[xx][yy])
        {
            visited[xx][yy][i][0] = true;
            CurrList.push(Node(xx,yy,i,0));
        }
    }
    while(!CurrList.empty())
    {
        Node u = CurrList.front();
        CurrList.pop();
        //cerr << u.x << ' ' << u.y << ' ' << u.dir << ' ' << u.turn << '\n';
        int xx = u.x + dx[u.dir];
        int yy = u.y + dy[u.dir];
        if (1 <= xx && xx <= n && 1 <= yy && yy <= m && !rip[xx][yy] &&
!visited[xx][yy][u.dir][u.turn])
        {
            visited[xx][yy][u.dir][u.turn] = true;
            CurrList.push(Node(xx,yy,u.dir,u.turn));
        }
        for (int i=0;i<4;i++)
            if (i != u.dir)
            {
                int xx = u.x + dx[i];
                int yy = u.y + dy[i];
                int tmp = 1;
                if (i == ((u.dir + 2)%4))
                    tmp = 0;
                if (1 <= xx && xx <= n && 1 <= yy && yy <= m &&
!rip[xx][yy] && !visited[xx][yy][i][tmp])
                {
                    visited[xx][yy][i][tmp] = true;
                    CurrList.push(Node(xx,yy,i,tmp));
                }
            }
    }
}

int main()
{

```

```

ios::sync_with_stdio(false);
cin.tie(0);
cout.tie(0);

freopen( (filename + ".inp").c_str(), "r", stdin);
freopen( (filename + ".out").c_str(), "w", stdout);

cin >> n >> m;
for (int i=1;i<=n;i++)
{
    string s;
    cin >> s;
    for (int j=0;j<m;j++)
    {
        a[i][j+1] = s[j];
        if (a[i][j+1] == 's')
        {
            s_x = i;
            s_y = j + 1;
        }
        else if (a[i][j+1] == 'f')
        {
            f_x = i;
            f_y = j + 1;
        }
        else if (a[i][j+1] == '#')
            rip[i][j+1] = true;
    }
}
bfs();
bool ok1 = false;
for (int i=0;i<4;i++)
    for (int j=0;j<2;j++)
        if (visited[f_x][f_y][i][j])
            ok1 = true;
if (!ok1)
{
    cout << -1;
    return 0;
}
bool ok2 = false;
for (int i=0;i<4;i++)
    if (visited[f_x][f_y][i][1])
        ok2 = true;
if (ok2)
{
    int p = abs(f_x - s_x) + abs(f_y - s_y);
    cout << (p%2);
}
else
    cout << abs(f_x - s_x) + abs(f_y - s_y);
}

```

## Bài 17: Thích đồ ngọt

Trong đêm đốt lửa trại các bạn nắm tay nhau múa tập thể vòng quanh đồng lửa. Khi điệu múa kết thúc Alice chạy đi và lát sau quay về với một hộp kẹo lớn, đựng  $k$  chiếc kẹo.

Trong vòng tròn còn lại  $n$  người, đánh số từ 1 đến  $n$  theo chiều kim đồng hồ, bắt đầu từ Tôm – người bạn thân của Alice. Alice đưa hộp kẹo cho bạn số  $lb$ . Hộp kẹo được chuyển tay nhau theo chiều kim đồng hồ. Mỗi người khi nhận hộp kẹo (kể cả bạn số  $lb$ ) lấy 1 hoặc 2 chiếc kẹo. Những người thích đồ ngọt lấy 2 chiếc, không thích – lấy một. Một người có thể được bốc nhiều lần vì hộp kẹo được chuyển đi vòng tròn.

Sau khi bốc hết kẹo người bốc viên kẹo cuối cùng trả hộp lại Alice và nếu Alice không nhầm thì đó là bạn số  $re$ . Sờ dĩ Alice quan tâm đến bạn cuối cùng vì có thể trong hộp khi nhận được chỉ còn 1 chiếc kẹo và dù bạn ấy thích đồ ngọt thì cũng chỉ có thể bốc một chiếc. Nghĩ rộng hơn, Alice tự hỏi không biết trong số các bạn nhận kẹo vừa rồi tối đa có bao nhiêu bạn thích đồ ngọt.

Hãy xác định số lượng nhiều nhất các bạn thích đồ ngọt trong hàng.

**Dữ liệu:** Vào từ file SWEET.INP gồm một dòng chứa 4 số nguyên  $n$ ,  $lb$ ,  $re$  và  $k$  ( $1 \leq n \leq 1000$ ,  $1 \leq lb, re \leq n$ ,  $1 \leq k \leq 10^9$ ).

**Kết quả:** Đưa ra file văn bản SWEET.OUT một số nguyên – số lượng nhiều nhất các bạn thích đồ ngọt. Nếu Alice nhầm lẫn – đưa ra số -1.

**Ví dụ:**

SWEET.INP	SWEET.OUT
5 3 4 10	3

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "SWEET";

#define int long long

int n, lb, re, k;
int cnt_a, cnt_b;
int res = -1;

int32_t main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    freopen( (filename + ".inp").c_str(), "r", stdin);
    freopen( (filename + ".out").c_str(), "w", stdout);

    cin >> n >> lb >> re >> k;
    cout << n;
    return 0;

    re = ((re - lb) + n) % n;
    lb = 0;
    for (int i = n; i >= 0; i--)
```

```

{
    int a = n + i;
    int b = k%a;
    if (b == 0)
        if (re == n-1)
        {
            res = i;
            break;
        }
    int mn, mx;
    if (re < n-i) mn = re + 1;
    else mn = re + 1 + (re - (n-i) + 1) - 1;
    if (re > i-1) mx = re + 1 + i;
    else mx = 2*(re+1);
    if (mn <= b && b <= mx)
    {
        res = i;
        break;
    }
}
cout << res;
}

```



## Bài 18: Sinh ba

Một gia đình có 3 con sinh ba. Các con còn nhỏ nên cái gì bố mẹ cũng phải chia đều cho 3 anh em. Hôm nay nhà trường tổ chức đi tham quan và đề nghị bố mẹ cho kẹo vào túi cho các cháu để học sinh có thể ăn trên đường đi.

Khi mở túi các con, mẹ thấy ở một túi còn **a** chiếc kẹo, túi thứ 2 còn **b** chiếc và túi còn lại – còn **c** chiếc. Mẹ rất vội vì sắp tới giờ đưa con đi học nên bà bốc kẹo cả 2 tay, bỏ đồng thời vào hai túi, mỗi túi một viên.

Hãy xác định số lần bỏ thêm kẹo ít nhất để có số kẹo ở ba túi bằng nhau.

**Dữ liệu:** Vào từ file TRINES.INP gồm một dòng chứa 3 số nguyên **a, b, c** ( $1 \leq a, b, c \leq 5 \times 10^8$ ).

**Kết quả:** Đưa ra file văn bản TRINES.OUT một số nguyên – số lần bỏ thêm kẹo tính được.

**Ví dụ:**

TRINES.INP	TRINES.OUT
1 2 3	3

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "TRINES";

#define int long long

int a[4];

int32_t main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    freopen( (filename + ".inp").c_str(), "r", stdin);
    freopen( (filename + ".out").c_str(), "w", stdout);

    cin >> a[1] >> a[2] >> a[3];
    sort(a+1, a+1+3);
    int res = 0;
    res += (a[2] - a[1]);
    res += (a[3] + (a[2] - a[1]) - a[2]);
    cout << res;
}
```

## Bài 19: Số đường đi

Cho lưới ô vuông kích thước  $n \times m$ , ô  $(1, 1)$  ở góc dưới trái, ô  $(n, m)$  – trên phải. Có  $k$  ô chứa chướng ngại vật, ô thứ  $i$  ở tọa độ  $(x_i, y_i)$ ,  $1 \leq x_i \leq n$ ,  $1 \leq y_i \leq m$ ,  $i = 1 \div k$ , Không có chướng ngại vật ở ô  $(1, 1)$  và  $(n, m)$ .

Rô bốt xuất phát từ ô  $(1, 1)$ , ở mỗi bước được chuyển sang ô kề cạnh bên phải hoặc bên trên nếu ô tới không chứa chướng ngại vật.

Hãy xác định số lượng đường rô bốt có thể đi từ ô  $(1, 1)$  đến ô  $(n, m)$  và đưa ra số lượng theo mô đun  $p$ , trong đó  $p$  – số nguyên tố.

**Dữ liệu:** Vào từ file văn bản ROUTES.INP:

- ✚ Dòng đầu tiên chứa 4 số nguyên  $n, m, k$  và  $p$  ( $1 \leq n, m \leq 10^5$ ,  $0 \leq k \leq 100$ ,  $2 \times \max\{m, n\} < p < 2 \times 10^9$ ),
- ✚ Nếu  $k > 0$ , dòng thứ  $i$  trong  $k$  dòng tiếp theo chứa 2 số nguyên  $x_i, y_i$  ( $1 \leq x_i \leq n$ ,  $1 \leq y_i \leq m$ ).

**Kết quả:** Đưa ra file văn bản ROUTES.OUT một số nguyên không âm – số lượng đường tìm được theo mô đun  $p$ .

**Ví dụ:**

ROUTES INP			
5	6	3	101
2	2		
3	5		
4	2		

ROUTES.OUT
25

```
#include <bits/stdc++.h>

using namespace std;
const string filename = "ROUTES";

#define int long long

int mod;

typedef pair<int,int> ii;
int h,w,n;
ii a[2001];
int d[2001];
int fact[200001];

void ext_e(int aa,int bb,int &x,int &y)
{
    if (bb==0)
    {
        x = 1;
        y = 0;
    }
}
```

```

        return;
    }
    int xx,yy;
    ext_e(bb,aa%bb,xx,yy);
    x = yy;
    y = xx - (aa / bb) * yy;
    return;
}

int inv(int aa)
{
    aa = ((aa%mod)+mod)%mod;
    int x,y;
    ext_e(aa,mod,x,y);
    return ((x%mod)+mod)%mod;
}

int comb(int k,int n)
{
    int cur = fact[n];
    cur = (cur * inv(fact[k]))%mod;
    cur = (cur * inv(fact[n-k]))%mod;
    return cur;
}

int32_t main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    freopen( (filename + ".inp").c_str(), "r", stdin);
    freopen( (filename + ".out").c_str(), "w", stdout);

    cin >> h >> w >> n >> mod;
    fact[0] = 1;
    for (int i=1;i<=200000;i++)
        fact[i] = (fact[i-1] * i)%mod;
    for (int i=1;i<=n;i++)
        cin >> a[i].first >> a[i].second;
    sort(a+1,a+1+n);
    for (int i=1;i<=n;i++)
    {
        int cur = comb(a[i].first-1,a[i].first+a[i].second-2);
        for (int j=1;j<i;j++)
            if (a[j].first <= a[i].first && a[j].second <= a[i].second)
                cur = ((cur - (d[j] * comb(a[i].first -
a[j].first,a[i].first + a[i].second - a[j].first - a[j].second))%mod)%mod +
mod)%mod;
        d[i] = cur;
    }
    int res = comb(h-1,h+w-2);
    for (int i=1;i<=n;i++)
    {
        int cur = d[i];
        cur = (cur * comb(h-a[i].first,h+w-a[i].first-a[i].second))%mod;
        res = ((res - cur)%mod + mod)%mod;
    }
}

```

```
    cout << res;  
}
```