```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef pair<ll, ll> ii;
typedef pair<double, double> dd;
typedef vector<ll> vi;
typedef vector<ii> vii;
int dr[] = { 0, 1, -1, 0, 1, -1, -1,  1};
int dc[] = {-1, 0,  0, 1, 1,  1, -1, -1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

main () {


}

```

```cpp
#include <bits/stdc++.h>
#define eps 1e-8
#define pi acos(-1)
using namespace std;
typedef long long ll;
double n, l, lo, hi, b, B, h, a;

double f (double x) {

    return pi * n * (a*a*x*x*x / 3.0 + a*b*x*x + x*b*b);
}

main () {

    int t; scanf ("%d", &t);
    while (t--) {

        scanf ("%lf %lf", &n, &l);
        scanf ("%lf %lf %lf", &b, &B, &h);
        a = (B-b) / h;
        lo = 0, hi = h;

        for (int i = 0; i < 100; ++i) {

            double mid = (hi+lo) / 2.0;

            if (f (mid) > l)
                hi = mid;
            else
                lo = mid;
        }

        printf ("%.2lf\n", lo);
    }
}

```

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

ll n, x;
string s;

ll character [100005], BIT[100005];

ll get sum (ll index) {

    ll sum = 0;

    while (index > 0) {

        sum += BIT[index];

        index -= index & (-index);
    }

    cout << sum << '\n';
}

void update (int index, int val) {

    index = index + 1;

    while (index <= n) {

       BIT[index] += val;

       index += index & (-index);
    }
}

void new BIT () {

    for (int i = 1; i <= n; ++i)
        BIT[i] = 0;

    for (int i = 0; i < n; ++i)
        update (i, character[i]);
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    cin >> n;
    for (int i = 0; i < n; ++i)
        cin >> character[i];

    new BIT ();

    while (cin >> s >> x) {

        --x;

        if (s == "a")
            update (x, -character[x]);

        else
            get sum (x);
    }
}
```

data structure - bitmask count bits.cpp

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef pair<ll, ll> ii;
typedef pair<double, double> dd;
typedef vector<ll> vi;
typedef vector<ii> vii;
int dr[] = { 0, 1, -1, 0, 1, -1, -1,  1};
int dc[] = {-1, 0,  0, 1, 1,  1, -1, -1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e4 + 10;
llu mask[N];

main () {

    ll t; scanf ("%lld", &t);
    while (t--) {

        ll n; scanf ("%lld", &n);
        for (ll i = 1; i <= n; ++i) {

            mask[i] = 0;
            ll m; scanf ("%lld", &m);
            while (m--) {

                ll x; scanf("%lld", &x);
                mask[i] |= (1LL << x);
            }
        }

        ll m; scanf ("%lld", &m);
        while (m--) {

            ll op, x, y; scanf ("%lld %lld %lld", &op, &x, &y);

            if (op == 1)
                printf ("%d\n",   builtin popcountll (mask[x] & mask[y]));

            else
                printf ("%d\n",   builtin popcountll (mask[x] | mask[y]));
        }
    }
}

```

```
#include <bits/stdc++.h>
#define pb push_back
#define mp make_pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef pair<ll, ll> ii;
typedef pair<double, double> dd;
typedef vector<ll> vi;
typedef vector<ii> vii;
int dr[] = { 0, 1, -1, 0, 1, -1, -1,  1};
int dc[] = {-1, 0,  0, 1, 1,  1, -1, -1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e4 + 10;
bitset<70> mask[N];

main () {

    ll t; scanf ("%lld", &t);
    while (t--) {

        ll n; scanf ("%lld", &n);
        for (ll i = 1; i <= n; ++i) {

            mask[i].reset ();
            ll m; scanf ("%lld", &m);
            while (m--) {

                ll x; scanf("%lld", &x);
                mask[i][x] = true;
            }
        }

        ll m; scanf ("%lld", &m);
        while (m--) {

            ll op, x, y; scanf ("%lld %lld %lld", &op, &x, &y);

            if (op == 1)
                printf ("%lu\n", (mask[x] & mask[y]).count ());

            else
                printf ("%lu\n", (mask[x] | mask[y]).count ());
        }
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;

bool flag = false;
set<int> numbers;
map<int, int> ans, l;
vector<int> pos;

main() {

    ios base::sync with stdio (0);
    cin.tie (0);

    int n, v;

    cin >> n >> v;
    numbers.insert(v);

    --n;
    while (n--) {

        cin >> v;
        auto it = numbers.upper bound(v);

        if (it != numbers.end() && l.count(*it) == 0)
            l[*it] = v;

        else
            it--;

        numbers.insert (v);
        pos.push back (v);
        ans[v] = *it;
    }

    cin >> n;
    while (n--) {

        cin >> v;

        if (flag)
            cout << ' ';

        else
            flag = true;

        cout << ans[pos[v-2]];
    }

    cout << '\n';
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

ll n, m;

char v[205][205];
ll sum[205][205];

void make () {

    for (ll i = 0; i <= n; ++i) {

        for (ll j = 0; j <= m; ++j) {

            if (!i || !j)
                sum[i][j] = 0;

            else
                sum[i][j] = v[i][j] + sum[i-1][j] + sum[i][j-1] - sum[i-1][j-1];
        }
    }
}

char solve (ll x) {

    for (ll i = 1; i <= n-x; ++i)
        for (ll j = 1; j <= m-x; ++j)
            if (sum[i+x][j+x] + sum[i-1][j-1] - sum[i-1][j+x] - sum[i+x][j-1] == 0)
                return true;

    return false;
}

main () {

    scanf ("%lld %lld", &n, &m);
    for (ll i = 1; i <= n; ++i) {

        for (ll j = 1; j <= m; ++j) {

            ll x; scanf ("%lld", &x);
            v[i][j] = !x;
        }
    }

    make ();

    ll t; scanf ("%lld", &t);
    while (t--) {

        ll x; scanf ("%lld", &x);
        puts (solve (x-1) ? "yes" : "no");
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

const ll N = 1e6;
ll st[N], lazy[N];
ll n;

void update (ll p, ll l, ll r, ll a, ll b, ll v) {

    if (lazy[p]!=0) {

        st[p] += (r-l+1) * lazy[p];

        if (l != r) {

            lazy[p<<1] += lazy[p];
            lazy[(p<<1)+1] += lazy[p];
        }

        lazy[p]=0;
    }

    if (r<a || l>b)
        return;

    if (a<=l && b>=r) {

        st[p] += (r-l+1) * v;

        if (l!=r) {

            lazy[p<<1] += v;
            lazy[(p<<1)+1] += v;
        }

        return;
    }

    ll mid = (l+r) >> 1;

    update (p<<1, l, mid, a, b, v);
    update ((p<<1)+1, mid+1, r, a, b, v);

    st[p] = st[p<<1] + st[(p<<1)+1];
}

ll query (ll p, ll l, ll r, ll a, ll b) {

    if (lazy[p]!=0) {

        st[p]+= (r-l+1) * lazy[p];

        if (l!=r) {

            lazy[p<<1] += lazy[p];
            lazy[(p<<1)+1] += lazy[p];
        }

        lazy[p]=0;
    }

    if (r<a || l>b)
        return 0;

    if (a<=l && b>=r)
        return st[p];

    ll mid = (r+l) >> 1;

    return query (p<<1, l, mid, a, b) + query ((p<<1)+1, mid+1, r, a, b);
}

main () {

    ll tc; scanf ("%lld", &tc);
    while (tc--){

        ll q; scanf ("%lld %lld", &n, &q);

        memset (st, 0, sizeof st);
        memset (lazy, 0, sizeof lazy);

        ll a, b, v;
        while (q--) {

            ll x; scanf ("%lld %lld %lld", &x, &a, &b);

            if (x)
                printf ("%lld\n", query (1, 1, n, a, b));

            else {

                scanf ("%lld", &v);
                update (1, 1, n, a, b, v);
            }
        }
    }
}

```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef pair<double, double> pdd;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

double dist (pdd a, pdd b) {

    return sqrt ((a.F-b.F)*(a.F-b.F) + (a.S-b.S)*(a.S-b.S));
}

const ll N = 2 * 1e3 + 10;
ll n;
pdd p0, points[N];
stack<pdd> s;

pdd nextToTop () {

    pdd p = s.top (); s.pop ();
    pdd res = s.top (); s.push (p);
    return res;
}

int orientation (pdd p, pdd q, pdd r) {

    double val = (q.S - p.S) * (r.F - q.F) - (q.F - p.F) * (r.S - q.S);

    if (val == 0)
        return 0;

    return (val > 0) ? 1 : 2;
}

int compare (pdd a, pdd b) {

    double o = orientation (p0, a, b);

    if (o == 0)
        return dist (p0, b) >= dist (p0, a);

    return o == 2;
}

void solve () {

    double ymin = points[0].S;
    ll min = 0;
    for (ll i = 1; i < n; ++i) {

        double y = points[i].S;
        if ((y < ymin) or (ymin == y and points[i].F < points[min].F))
            ymin = points[i].S, min = i;
    }

    swap (points[0], points[min]);
    p0 = points[0];
    sort (points+1, points+n, compare);

    ll m = 1;
    for (ll i = 1; i < n; ++i) {

        while (i < n-1 and orientation (p0, points[i], points[i+1]) == 0)
            i++;

        points[m++] = points[i];
    }

    s.push (points[0]); s.push (points[1]); s.push (points[2]);
    for (ll i = 3; i < m; i++) {

        while (orientation (nextToTop (), s.top (), points[i]) != 2)
            s.pop ();

        s.push (points[i]);
    }

    double ans = 0;
    pdd fst = s.top ();
    pdd before = s.top(); s.pop();
    while (!s.empty()) {

        ans += dist (before, s.top());
        before = s.top (); s.pop ();
    }

    ans += dist (before, fst);
    printf ("Tera que comprar uma fita de tamanho %.2lf.\n", ans);
}

void read () {

    for (ll i = 0; i < n; ++i)
        scanf ("%lf %lf", &points[i].F, &points[i].S);
}

main () {

    while (scanf ("%lld", &n), n) {

        read ();
        solve ();
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
typedef unsigned long long int ull;
typedef pair<int,int> pii;
const int MAX = 51;

int n, m;

char board 1[MAX][MAX], board 2[MAX][MAX];
char visited [MAX][MAX][MAX][MAX];

bool validPosition (int x1, int y1, int x2, int y2) {

    if (x1 < 0 || x1 == n || x2 < 0 || x2 == n)
        return false;

    if (y1 < 0 || y1 == m || y2 < 0 || y2 == m)
        return false;

    if (board 1[x1][y1] == 'B' || board 2[x2][y2] == 'B')
        return false;

    return true;
}

void bfs (pii source 1, pii source 2, pii destination 1, pii destination 2) {

    memset (visited, false, sizeof visited);
    queue< pair<pair<pii, pii>, int> > q;

    q.push ({{source 1, source 2}, 0});
    visited [source 1.first][source 1.second][source 2.first][source 2.second] = true;

    while (!q.empty ()) {

        int x1 = q.front ().first.first.first;
        int y1 = q.front ().first.first.second;
        int x2 = q.front ().first.second.first;
        int y2 = q.front ().first.second.second;
        int moves = q.front ().second;
        q.pop ();

        if (moves == MAX) {

            printf ("impossivel\n");
            return;
        }

        if (x1 == destination 1.first && y1 == destination 1.second && x2 == destination 2.first && y2 == destination 2.second) {

            printf ("%d\n", moves);
            return;
        }

        // down
        if (validPosition (x1+1, y1, x2+1, y2)) {

            if (board 1[x1+1][y1] == '#' && board 2[x2+1][y2] != '#') {

                if (visited[x1][y1][x2+1][y2] == false) {

                    q.push ({{{x1, y1}, {x2+1, y2}}, moves+1});
                    visited[x1][y1][x2+1][y2] = true;
                }
            }

            if (board 1[x1+1][y1] != '#' && board 2[x2+1][y2] == '#') {

                if (visited[x1+1][y1][x2][y2] == false) {

                    q.push ({{{x1+1, y1}, {x2, y2}}, moves+1});
                    visited[x1+1][y1][x2][y2] = true;
                }
            }

            if (board 1[x1+1][y1] != '#' && board 2[x2+1][y2] != '#') {

                if (visited[x1+1][y1][x2+1][y2] == false) {

                    q.push ({{{x1+1, y1}, {x2+1, y2}}, moves+1});
                    visited[x1+1][y1][x2+1][y2] = true;
                }
            }
        }

        // up
        if (validPosition (x1-1, y1, x2-1, y2)) {

            if (board 1[x1-1][y1] == '#' && board 2[x2-1][y2] != '#') {

                if (visited[x1][y1][x2-1][y2] == false) {

                    q.push ({{{x1, y1}, {x2-1, y2}}, moves+1});
                    visited[x1][y1][x2-1][y2] = true;
                }
            }

            if (board 1[x1-1][y1] != '#' && board 2[x2-1][y2] == '#') {

                if (visited[x1-1][y1][x2][y2] == false) {

                    q.push ({{{x1-1, y1}, {x2, y2}}, moves+1});
                    visited[x1-1][y1][x2][y2] = true;
                }
            }

            if (board 1[x1-1][y1] != '#' && board 2[x2-1][y2] != '#') {

                if (visited[x1-1][y1][x2-1][y2] == false) {

                    q.push ({{{x1-1, y1}, {x2-1, y2}}, moves+1});
                    visited[x1-1][y1][x2-1][y2] = true;
                }
            }
        }

        // left
        if (validPosition (x1, y1-1, x2, y2-1)) {

            if (board 1[x1][y1-1] == '#' && board 2[x2][y2-1] != '#') {

                if (visited[x1][y1][x2][y2-1] == false) {

                    q.push ({{{x1, y1}, {x2, y2-1}}, moves+1});
                    visited[x1][y1][x2][y2-1] = true;
                }
            }

            if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] == '#') {

                if (visited[x1][y1-1][x2][y2] == false) {

                    q.push ({{{x1, y1-1}, {x2, y2}}, moves+1});
                    visited[x1][y1-1][x2][y2] = true;
                }
            }

            if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] != '#') {

                if (visited[x1][y1-1][x2][y2-1] == false) {

                    q.push ({{{x1, y1-1}, {x2, y2-1}}, moves+1});
                    visited[x1][y1-1][x2][y2-1] = true;
                }
            }
        }

        // right
        if (validPosition (x1, y1+1, x2, y2+1)) {

            if (board 1[x1][y1+1] == '#' && board 2[x2][y2+1] != '#') {

                if (visited[x1][y1][x2][y2+1] == false) {

                    q.push ({{{x1, y1}, {x2, y2+1}}, moves+1});
                    visited[x1][y1][x2][y2+1] = true;
                }
            }

            if (board 1[x1][y1+1] != '#' && board 2[x2][y2+1] == '#') {

                if (visited[x1][y1+1][x2][y2] == false) {

                    q.push ({{{x1, y1+1}, {x2, y2}}, moves+1});
                    visited[x1][y1+1][x2][y2] = true;
                }
            }

            if (board 1[x1][y1+1] != '#' && board 2[x2][y2+1] != '#') {

                if (visited[x1][y1+1][x2][y2+1] == false) {

                    q.push ({{{x1, y1+1}, {x2, y2+1}}, moves+1});
                    visited[x1][y1+1][x2][y2+1] = true;
                }
            }
        }
    }
}

main () {

    int t;
    scanf ("%d", &t);
    while (t--) {

        scanf ("%d %d", &n, &m);

        pii source 1, destination 1;

        for (int i = 0; i < n; ++i) {
            getchar ();
            for (int j = 0; j < m; ++j) {

                scanf ("%c", &board 1[i][j]);

                if (board 1[i][j] == 'R')
                    source 1 = {i, j};

                else if (board 1[i][j] == 'F')
                    destination 1 = {i, j};
            }
        }

        pii source 2, destination 2;

        for (int i = 0; i < n; ++i) {
            getchar ();
            for (int j = 0; j < m; ++j) {

                scanf ("%c", &board 2[i][j]);

                if (board 2[i][j] == 'R')
                    source 2 = {i, j};

                else if (board 2[i][j] == 'F')
                    destination 2 = {i, j};
            }
        }

        bfs (source 1, source 2, destination 1, destination 2);
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
#define MAX 1003

int n;

bool g [MAX][MAX];
char visited [MAX];

bool bfs (int source) {

    queue< pair<int, int> > q;
    memset (visited, 0, sizeof visited);

    visited[source] = 1;
    q.push (make pair (source, 2));

    while (!q.empty ()) {

        int i = q.front ().first;
        int color = q.front ().second;
        q.pop ();

        for (int j = 0; j < n; ++j) {

            if (g[i][j] == false) {

                if (visited[j] == 0)
                    q.push (make pair (j, (color == 1) ? 2 : 1)), visited[j] = color;

                else if (visited[j] != color)
                    return false;
            }
        }
    }

    return true;
}

main () {

    scanf ("%d", &n);

    for (int i = 0; i < n; ++i)
        for (int j = 0; j < n; ++j)
            scanf ("%d", &g[i][j]);

    puts (bfs (0) ? "Bazinga!" : "Fail!");
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

vector<int> g [10005];
int visited [10005];
bool loop;

void dfs (int i) {

    visited[i] = 1;

    for (int j = 0; loop == false && j < g[i].size (); ++j) {

        if (visited[g[i][j]] == 0)
            dfs (g[i][j]);

        else if (visited[g[i][j]] == 1)
            loop = true;
    }

    visited[i] = 2;
}

main () {

    ios_base::sync_with_stdio (0);
    cin.tie (0);

    int i, x, y, t, n, m;

    cin >> t;
    while (t--) {

        cin >> n >> m;

        for (i = 1; i <= n; ++i)
            g[i].clear ();

        while (m--) {

            cin >> x >> y;
            g[x].emplace_back (y);
        }

        loop = false;
        memset (visited, 0, sizeof (visited));

        for (i = 1; loop == false && i <= n; ++i)
            if (visited[i] != 1)
                dfs (i);

        puts (loop ? "SIM" : "NAO");
    }
}

```

```cpp
#include <bits/stdc++.h>
#define eoq cout << "eoq" << endl
#define F first
#define S second
using namespace std;
typedef long long int ll;
typedef long long unsigned int llu;
typedef pair<int, int> pii;
const int N = 1e4 + 1;

ll n, vertex, weigth;

vector<ll> g[N];
char visited[N];

void reset () {

    for (ll i = 1; i <= n; ++i)
        g[i].clear ();
}

void read () {

    for (ll i = 2; i <= n; ++i) {

        ll j;
        scanf ("%lld", &j);
        g[i].push_back (j);
        g[j].push_back (i);
    }
}

void bfs (int i, int wt) {

    memset (visited, false, sizeof visited);
    visited[i] = true;

    queue<pii> q;
    q.push (make_pair (i, wt));

    while (!q.empty ()) {

        i = q.front ().F;
        wt = q.front ().S;
        q.pop ();

        for (int j = 0; j < g[i].size (); ++j) {

            if (visited[g[i][j]] == false) {

                visited[g[i][j]] = true;
                q.push (make_pair (g[i][j], wt+1));
            }
        }
    }

    vertex = i;
    weigth = wt;
}

void solve () {

    vertex = weigth = 0;
    bfs (1, 0);
    bfs (vertex, 0);
    printf ("%lld\n", weigth/2 + (weigth & 1));
}

main () {

    while (scanf ("%lld", &n), n != -1) {

        reset ();
        read ();
        solve ();
    }
}

```

graph - dijkstra.cpp
Page 1 of 1

```cpp
#include <bits/stdc++.h>
using namespace std;
#define pii pair<int,int>

int cost[26][26];
int minimum;
char visited[26];

int bfs (int source, int destination) {

    priority queue <pii, vector<pii>, greater<pii> > pq;
    memset (visited, false, sizeof visited);

    pq.push (make pair (0, source));

    while (!pq.empty ()) {

        int qtd = pq.top ().first;
        int i = pq.top ().second;

        if (i == destination || qtd >= minimum)
            return qtd;

        pq.pop ();
        visited[i] = true;
        // mark visited after removing from queue

        for (int j = 0; j < 26; ++j)
            if (visited[j] == false)
                pq.push (make pair (qtd+cost[i][j], j));
    }
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    for (int i = 0; i < 26; ++i)
        for (int j = 0; j < 26; ++j)
            cin >> cost[i][j];

    string s;
    cin >> s;

    int idx1 = 0;
    int idx2 = s.length () - 1;
    int answer = 0;

    while (idx1 < idx2) {

        if (s[idx1] != s[idx2]) {

            minimum = 1e6;

            for (int i = 0; i < 26; ++i)
                minimum = min (minimum, bfs (s[idx1] - 'a', i) + bfs (s[idx2] - 'a', i));

            answer += minimum;
        }

        ++idx1;
        --idx2;
    }

    cout << answer << '\n';
}
```

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

char g[1025][1025];
int n, m, answer;

void dfs (int i, int j) {

    ++g[i][j];

    if (i > 0 && g[i-1][j] == '.')
        dfs (i-1, j);

    if (i < n-1 && g[i+1][j] == '.')
        dfs (i+1, j);

    if (j > 0 && g[i][j-1] == '.')
        dfs (i, j-1);

    if (j < m-1 && g[i][j+1] == '.')
        dfs (i, j+1);
};

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    cin >> n >> m;

    for (int i = 0; i < n; ++i)
        cin >> g[i];

    answer = 0;
    for (int i = 0; i < n; ++i) {

        for (int j = 0; j < m; ++j) {

            if (g[i][j] == '.') {

                ++answer;
                dfs (i, j);
            }
        }
    }

    cout << answer << endl;
}

```

```
#include <bits/stdc++.h>
using namespace std;
typedef long long unsigned int llu;

llu cost[26][26];

void floydWarshall () {

    for (llu k = 0; k < 26; k++)
        for (llu i = 0; i < 26; i++)
            for (llu j = 0; j < 26; j++)
                cost[i][j] = min (cost[i][j], cost[i][k] + cost[k][j]);
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    for (llu i = 0; i < 26; ++i)
        for (llu j = 0; j < 26; ++j)
            cin >> cost[i][j];

    for (llu i = 0; i < 26; ++i)
        cost[i][i] = 0;

    string s;
    cin >> s;

    llu idx1 = 0;
    llu idx2 = s.length () - 1;
    llu answer = 0;

    floydWarshall ();

    while (idx1 < idx2) {

        if (s[idx1] != s[idx2]) {

            llu minimum = 1e15;

            for (llu i = 0; i < 26; ++i)
                minimum = min (minimum,  cost[s[idx1] - 'a'][i] + cost[s[idx2] - 'a'][i]);

            answer += minimum;
        }

        ++idx1;
        --idx2;
    }

    cout << answer << '\n';
}

```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 100;
map<string, ll> string to num;
string num to string[N];
vector<ll> g1[N], g2[N];
char visited[N];
stack<ll> s;
ll n, m, c = 1;
bool flag, endline = false;

void reset () {

    for (ll i = 0; i <= n; ++i)
        g1[i].clear (), g2[i].clear ();

    string to num.clear ();
}

void read () {

    ll x, y, total = 0;
    string a, b;

    while (m--) {

        cin >> a >> b;

        if (!string to num.count (a))
            string to num[a] = total, num to string[total] = a, total++;

        if (!string to num.count (b))
            string to num[b] = total, num to string[total] = b, total++;

        x = string to num[a];
        y = string to num[b];

        g1[x].pb (y);
        g2[y].pb (x);
    }
}

void dfs1 (ll i) {

    visited[i] = true;

    for (ll j = 0; j < g1[i].size (); ++j)
        if (!visited[g1[i][j]])
            dfs1 (g1[i][j]);

    s.push (i);
}

void dfs2 (ll i) {

    if (flag == true)
        printf (", ");

    else
        flag = true;

    cout << num to string[i];

    visited[i] = true;
    for (ll j = 0; j < g2[i].size (); ++j)
        if (!visited[g2[i][j]])
            dfs2 (g2[i][j]);
}

void solve () {

    memset (visited, false, sizeof visited);
    for (ll i = 0; i < n; ++i)
        if (!visited[i])
            dfs1 (i);

    ll current = 0LL; flag = false;
    memset (visited, false, sizeof visited);

    if (endline)
        printf ("\n");

    else
        endline = true;

    printf ("Calling circles for data set %lld:\n", c++);

    while (!s.empty ()) {

        ll i = s.top (); s.pop ();
        if (!visited[i]) {

            flag = false;
            dfs2 (i);
            printf ("\n");
        }
    }
}

main () {

    while (scanf ("%lld %lld", &n, &m), n and m) {

        reset ();
        read ();
        solve ();
    }
}

```

graph - kruskal.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

const ll N = 1e5 + 10;
vector< pair< ll, pair<ll, ll> > > g;
ll parent[N];
ll n, m, total;

void reset () {

    for (int i = 0; i <= n; ++i)
        parent[i] = i;

    g.clear ();
    total = 0;
}

ll findset (ll x) {

    if (x != parent[x])
        parent[x] = findset (parent[x]);

    return parent[x];
}

void UNION (ll x, ll y) {

    parent[x] = parent[y];
}

void solve () {

    ll pu, pv, total = 0;

    sort (g.begin (), g.end ());

    for (ll i = 0; i < g.size (); i++) {

        pu = findset (g[i].second.first);
        pv = findset (g[i].second.second);

        if (pu != pv) {

            total += g[i].first;
            UNION (pu, pv);
        }
    }

    printf ("%lld\n", total);
}

void read () {

    while (m--) {

        ll x, y, w; scanf ("%lld %lld %lld", &x, &y, &w);
        g.push_back (make_pair (w, make_pair (x, y)));
    }
}

main () {

    scanf ("%lld %lld", &n, &m);
    reset ();
    read ();
    solve ();
}

```

math - base conversion.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;

string str;
void convert (int n, int base) {

    if (n >= base)
        convert ((n / base), base);

    char c = n % base;
    if (c > 9)
        str += (c + 'a' - 10);

    else
        str += (c + '0');
}

main () {

    int t, c = 1;
    string n, b;

    cin >> t;
    while (t--) {

        cin >> n >> b;

        cout << "Case " << c++ << ":\n";

        if (b == "bin") {

            str.clear ();
            convert (stoi (n, nullptr, 2), 10);
            cout << str << " dec\n";

            str.clear ();
            convert (stoi (n, nullptr, 2), 16);
            cout << str << " hex\n";
        }

        else if (b == "dec") {

            str.clear ();
            convert (stoi (n, nullptr, 10), 16);
            cout << str << " hex\n";

            str.clear ();
            convert (stoi (n, nullptr, 10), 2);
            cout << str << " bin\n";
        }

        else if (b == "hex") {

            str.clear ();
            convert (stoi (n, nullptr, 16), 10);
            cout << str << " dec\n";

            str.clear ();
            convert (stoi (n, nullptr, 16), 2);
            cout << str << " bin\n";
        }

        cout << '\n';
    }
}

```

math - factorial (last digit).cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

ll convert (ll n) {

    if (n == 0)
        return 1;

    if (n == 1)
        return 1;

    if (n == 2)
        return 2;

    if (n == 3)
        return 6;

    if (n == 4)
        return 4;

    if (n == 5)
        return 2;

    if (n == 6)
        return 2;

    if (n == 7)
        return 4;

    if (n == 8)
        return 2;

    if (n == 9)
        return 8;

    if (n == 10)
        return 8;

    if ((n/10) % 2 == 0)
        return 6 * convert (n/5) * convert (n%10);

    else
        return 4 * convert (n/5) * convert (n%10);
}

main () {

    ll n;
    int c = 1;

    while (scanf ("%lld", &n) != EOF)
        printf ("Instancia %d\n%lld\n\n", c++, convert (n) % 10);
}

```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

main () {

    ll n; scanf ("%lld", &n);

    if (n == 1 or n == 2)
        printf ("-1\n");

    else if (n & 1)
        printf ("%lld %lld\n", (n*n-1) >> 1, (n*n+1) >> 1);

    else
        printf ("%lld %lld\n", (n >> 1) * (n >> 1) - 1, (n >> 1) * (n >> 1) + 1);
}
```

```
#include <bits/stdc++.h>
using namespace std;
typedef pair<int,int> ii;
typedef vector<ii> perm t;

#define PERM SZ 21
int cw[PERM SZ]= { 13,8,3,18,14,9,4,0,19,15,10,5,1,20,16,11,6,2,17,12,7 }; // clock wise perm.
int ccw[PERM SZ] = { 7,12,17,2,6,11,16,20,1,5,10,15,19,0,4,9,14,18,3,8,13 }; // counter clock wise perm.
// permutable faces with its affeted values
int f[PERM SZ] = { 16,17,18,3,19,20,21,37,6,22,23,24,40,9,25,26,27,43,28,29,30 };
int r[PERM SZ] = { 18,15,12,21,37,38,39,46,24,40,41,42,49,27,43,44,45,52,30,33,36 };
int l[PERM SZ] = { 10,13,16,48,1,2,3,19,51,4,5,6,22,54,7,8,9,25,34,31,28 };
int b[PERM SZ] = { 12,11,10,39,46,47,48,1,42,49,50,51,4,45,52,53,54,7,36,35,34 };
int u[PERM SZ] = { 48,47,46,1,10,11,12,39,2,13,14,15,38,3,16,17,18,37,19,20,21 };
int d[PERM SZ] = { 25,26,27,9,28,29,30,43,8,31,32,33,44,7,34,35,36,45,54,53,52 };
// do-able moves (aka permutations) for the rubik cube
map<char,perm t> moves;

perm t makePerm (int *values, int *order) {

    perm t ret (PERM SZ);

    for (size t i = 0; i < PERM SZ; ++i)
        ret[i].first = values[i], ret[i].second = values[ order[i] ];

    return ret;
}

vector<int> getCubePerm (const char *seq) {

    vector<int> ret(55), tmp(55);

    for (size t i = 0; i < ret.size (); ++i)
        ret[i] = i;

    for (size t len = strlen (seq), i = 0; i < len; ++i) {

        const perm t& p = moves[seq[i]];

        for (size t j = 0; j < ret.size (); ++j)
            tmp[j] = ret[j]; // back up

        for (size t j = 0; j < p.size (); ++j)
            ret[ p[j].first ] = tmp[ p[j].second ]; // then permute
    }

    return ret;
}

int gcd (int a, int b) {

    int r;

    while (b > 0)
        r = a%b, a = b, b = r;

    return a;
}

inline int lcm (int a, int b) {

    return ((a / gcd(a,b)) * b);
}

int gcdOfCyclesLen (const vector<int> &p) {

    int ret = 1, cnt = 0;
    vector<bool> used (0, p.size ());
    size_t i, j;
```

```cpp

    for (i = 1; i < p.size (); ++i) {

        if (!used[i]) {

            used[i] = true, cnt++, j = p[i];

            while (j != i)
                used[j] = true, cnt++, j = p[j];

            ret = lcm (cnt, ret), cnt = 0;
        }
    }

    return ret;
}

main () {

    moves['U'] = makePerm(u,cw), moves['u'] = makePerm(u,ccw);
    moves['F'] = makePerm(f,cw), moves['f'] = makePerm(f,ccw);
    moves['D'] = makePerm(d,cw), moves['d'] = makePerm(d,ccw);
    moves['R'] = makePerm(r,cw), moves['r'] = makePerm(r,ccw);
    moves['L'] = makePerm(l,cw), moves['l'] = makePerm(l,ccw);
    moves['B'] = makePerm(b,cw), moves['b'] = makePerm(b,ccw);

    char input[85];
    while (gets(input) != NULL)
        printf ("%d\n", gcdOfCyclesLen (getCubePerm (input)));
}

```

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

typedef unsigned long long int lli;

lli i, j, k, lim = 20000000;

std::vector<bool> isprime (20000000, true);
std::vector<bool> v;

void primes () {

    for (i = 2; i < lim; ++i)
        if (isprime[i])
            for (j = i*i; j < lim; j += i)
                isprime[j] = false;
}

main () {

    primes ();

    lli t, n, elem, i;

    scanf ("%llu", &t);

    for (; t; --t) {

        scanf ("%llu", &n);

        v = isprime;

        for (i = 0; n; --n) {

            scanf ("%llu", &elem);

            if (elem < lim)
                v[elem] = false;
        }

        for (i = 1; i < lim; ++i)
            if (v[i])
                break;

        printf("%llu\n", i-1);
    }
}

```

```
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};
ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

ll i, j;

ll end (ll n, ll i) {

    if (i == (n >> 1) + (n & 1))
        return n*n;

    ll a1 = (n << 2) - 4;
    ll r = -8;
    ll ai = a1 + (i-1) * r;

    return ((a1 + ai) * i) >> 1;
}

ll begin (ll n, ll i) {

    if (i == 1)
        return 1;

    return end (n, i-1) + 1;
}

ll go right (ll n, ll layer, ll pos, ll current) {

    ll aux = current + (n - ((layer-1) << 1)) - 1;
    j += min (aux, pos) - current;
    return aux;
}

ll go down (ll n, ll layer, ll pos, ll current) {

    ll aux = current + (n - ((layer-1) << 1)) - 1;
    i += min (aux, pos) - current;
    return aux;
}

ll go left (ll n, ll layer, ll pos, ll current) {

    ll aux = current + (n - ((layer-1) << 1)) - 1;
    j -= min (aux, pos) - current;
    return aux;
}

ll go up (ll n, ll layer, ll pos, ll current) {

```

```cpp
    ll aux = current + (n - ((layer-1) << 1)) - 2;
    i -= min (aux, pos) - current;
    return aux;
}

void solve (ll n, ll layer, ll pos, ll idx begin) {

    i = j = layer;
    ll current = idx begin;
    if (current < pos)
        current = go right (n, layer, pos, current);

    if (current < pos)
        current = go down (n, layer, pos, current);

    if (current < pos)
        current = go left (n, layer, pos, current);

    if (current < pos)
        current = go up (n, layer, pos, current);

    printf ("%lld %lld\n", i, j);
}

main () {

    ll n, pos; scanf ("%lld %lld", &n, &pos);
    ll lo = 1; // first layer
    ll hi = (n >> 1) + (n & 1); // last layer

    while (true) {

        ll mid = (lo + hi) >> 1;
        ll idx begin = begin (n, mid);
        ll idx end = end (n, mid);

        if (pos >= idx begin and pos <= idx end) {

            solve (n, mid, pos, idx begin);
            break;
        }

        if (pos > idx end)
            lo = mid+1;

        if (pos < idx begin)
            hi = mid;
    }
}

```

paradigms - baloons (top down).cpp

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef pair<double, double> pdd;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e2 + 10;
ll dp[N][N][10], value[N], n;
char s[N];

ll solve (ll current, ll before, ll exponent) {

    if (current == n)
        return 0LL;

    if (dp[current][before][exponent] != -1)
        return dp[current][before][exponent];

    ll ans = 0;
    ll pt = value[s[current] - 'a'];

    if (s[current] == s[before]) {

        pt = min (pt*(ll)pow(2, exponent), pt * 512);
        ans = max (solve (current+1, before, exponent), pt + solve (current+1, current, min (exponent+1, 9LL)));
    }

    else
        ans = max (solve (current+1, before, exponent), pt + solve (current+1, current, 1));

    return dp[current][before][exponent] = ans;
}

main () {

    ll t; scanf ("%lld", &t);
    while (t--) {

        for (ll i = 0; i < 3; ++i)
            scanf ("%lld", value + i);

        scanf ("%s", s); n = strlen (s);
        memset (dp, -1, sizeof dp);
        printf ("%lld\n", solve (0, 0, 0));
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

int n, total;

int coin [30], T[10000001], R[10000001];

void coinChange () {

    T[0] = 0;

    for (int i = 1; i <= total; ++i) {

        T[i] = INT MAX;
        R[i] = -1;
    }

    for (int j = 0; j < n; ++j) {

        for (int i = 1; i <= total; ++i) {

            if (i >= coin[j]) {

                if (T[i - coin[j]] + 1 < T[i]) {

                    T[i] = 1 + T[i - coin[j]];
                    R[i] = j;
                }
            }
        }
    }

    cout << T[total] << '\n';
}

// discover used coins
void print coinChange () {

    int i = total;

    while (i != 0) {

        int j = R[i];
        cout << coin[j] << '\n';
        i -= coin[j];
    }
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    int t;
    cin >> t;
    while (t--) {

        cin >> n >> total;

        for (int i = 0; i < n; ++i)
            cin >> coin[i];

        coinChange ();
    }
}
```

paradigms - coin change (top down).cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
#define MAX COIN 30
#define MAX TOTAL 1000006

int n;

int coin [MAX COIN], dp [MAX TOTAL];

int coinChange (int total, int idx) {

    if (total == 0)
        return 0;

    if (idx == n || total < 0)
        return MAX TOTAL;

    if (dp[total] != -1)
        return dp[total];

    // coinChange with repetition
    dp[total] = min (coinChange (total, idx+1), 1 + coinChange (total-coin[idx], idx));

    // coinChange 1-0
    //dp[total] = min (coinChange (total, idx+1), min (1 + coinChange (total-moeda[idx], idx+1), 1 + coinChange(total-moeda[idx], idx)));

    return dp[total];
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    int t;
    cin >> t;
    while (t--) {

        int total;
        cin >> n >> total;

        for (int i = 0; i < n; ++i)
            cin >> coin[i];

        for (int i = 0; i <= total; ++i)
            dp[i] = -1;

        cout << coinChange (total, 0) << '\n';
    }
}

```

```
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e3 + 10;
ii v[N];
ll n, max elem, min elem;

ll calculate (ll line, ll i) {

    ll ans = 1; ii before = v[i++];
    for (; i < n; ++i) // search points on curve
        if (v[i].F != before.F and v[i].S != before.S and abs (v[i].S - line) == 1) // new point on line
            ++ans, before = v[i];
    return ans;
}

ll get max (ll line) {

    ll i;
    for (i = 0; i < n and (abs (v[i].S - line) != 1); ++i); // search first point
    if (i == n)
        return 0;

    ll ans = calculate (line, i++);
    for (; i < n and v[i].F == v[i-1].F; ++i) // search all first points with same x
        if (abs (v[i].S - line) == 1)
            ans = max (ans, calculate (line, i));
    return ans;
}

void solve () {

    ll ans = 0;
    for (ll line = min elem + 1; line < max elem; ++line)
        ans = max (ans, get max (line));
    cout << ans << '\n';
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    while (cin >> n) {

        min_elem = LLONG_MAX;
        max elem = LLONG MIN;

        for (ll i = 0; i < n; ++i) {

            cin >> v[i].F >> v[i].S;
            min elem = min (min elem, v[i].S);
            max elem = max (max elem, v[i].S);
        }

        sort (v, v + n);
        solve ();
    }
}
```

paradigms - dp with fibonacci matrix exponentiation.cpp

```cpp
#include <bits/stdc++.h>
#define eps 1e-8
#define maxn 1000100
#define mod 1000000LL
using namespace std;

typedef long long ll;

ll n, m, o;

struct mat {

    ll v[2][2];

    mat () {

        memset(v, 0, sizeof v);
    }

    void init () {

        v[0][0] = v[1][1] = 1;
    }

    mat operator * (mat other) {

        mat res;
        for (int i = 0; i < 2; ++i)
            for (int j = 0; j < 2; ++j)
                for (int k = 0; k < 2; ++k)
                    res.v[i][j] = (res.v[i][j] + v[i][k]*other.v[k][j]) % mod;

        return res;
    }
};

main () {

    while (~scanf ("%lld %lld %lld", &n, &m, &o)) {

        m %= mod;
        o %= mod;
        n /= 5;
        mat ans, aux;
        ans.init ();
        aux.v[0][1] = 1;
        aux.v[1][0] = o;
        aux.v[1][1] = m;
        while (n > 0) {

            if (n & 1) ans = ans * aux;
            aux = aux*aux;
            n /= 2;
        }

        printf ("%06lld\n", (ans.v[0][1] * m + ans.v[0][0]) % mod);
    }
}

```

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
#define W 50

int n;

int wt[1005], val[1005], dp[1005][W+1];

void knapSack () {

    for (int i = 0; i <= n; i++) {

        for (int j = 0; j <= W; j++) {

            if (i==0 || j==0)
                dp[i][j] = 0;

            // knapsack 0-1
            else if (wt[i-1] <= j)
                dp[i][j] = max (dp[i-1][j], val[i-1] + dp[i-1][j-wt[i-1]]);

            // knapsack with repetition
                // dp[i][j] = max (dp[i-1][j], val[i-1] + dp[i][j-wt[i-1]]);

            else
                dp[i][j] = dp[i-1][j];
        }
    }

    // discover used val and wt
    int total b = 0;
    int total w = 0;
    int j = W;

    for (int i = n; i > 0; --i) {

        if (dp[i][j] != dp[i-1][j]) {

            total w += wt[i-1];
            ++total b;
            j -= wt[i-1];
        }
    }

    printf ("%d brinquedos\n", dp[n][W]);
    printf ("Peso: %d kg\n", total w);
    printf ("sobra(m) %d pacote(s)\n\n", n - total b);
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    int t;
    cin >> t;
    while (t--) {

        cin >> n;
        for (int i = 0; i < n; ++i)
            cin >> val[i] >> wt[i];

        knapSack ();
    }
}
```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e5 + 10;
ll hist[N], n;

void read () {

    scanf ("%lld", &n);
    for (ll i = 0; i < n; ++i)
        scanf ("%lld", &hist[i]);
}

void solve () {

    stack<ll> s;
    ll max area = 0;
    ll tp;
    ll area with top;

    ll i = 0;
    while (i < n) {

        if (s.empty () or hist[s.top ()] <= hist[i])
            s.push (i++);

        else {

            tp = s.top ();
            s.pop ();
            area with top = hist[tp] * (s.empty () ? i : i - s.top () - 1);

            if (max area < area with top)
                max area = area with top;
        }
    }

    while (!s.empty ()) {

        tp = s.top ();
        s.pop ();
        area with top = hist[tp] * (s.empty () ? i : i - s.top () - 1);

        if (max area < area with top)
            max area = area with top;
    }

    printf ("%lld\n", max area);
}

main () {

    while (true) {

        read ();

        if (!n)
            break;

        solve ();
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

int n, c, maxDiff;

main () {

    ios base::sync with stdio (0);
    cin.tie (0);

    cin >> n >> c;

    int prices [n], dp [n];

    for (int i = 0; i < n; ++i)
        cin >> prices[i];

    dp[0] = 0;
    maxDiff = -prices[0];

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

        dp[i] = max (dp[i - 1], prices[i-1] + maxDiff - c);
        maxDiff = max (maxDiff, dp[i-1] - prices[i-1]);
    }

    cout << dp[n] << endl;
}

```

paradigms - max profit (top down).cpp

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef pair<ll, ll> ii;
typedef pair<double, double> dd;
typedef vector<ll> vi;
typedef vector<ii> vii;
int dr[] = { 0, 1, -1, 0, 1, -1, -1,  1};
int dc[] = {-1, 0,  0, 1, 1,  1, -1, -1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 2 *1e5 + 10;
ll dp[N][2], v[N];
ll n, c;

void read () {

    scanf ("%lld %lld", &n, &c);

    for (ll i = 0; i < n; ++i)
        scanf ("%lld", v+i);

    memset (dp, -1, sizeof dp);
}

ll solve (ll current, ll has) {

    if (current == n)
        return 0LL;

    if (dp[current][has] != -1)
        return dp[current][has];

    if (has == 1)
        return dp[current][has] = max (solve (current+1, 0) + v[current], solve (current+1, 1));

    else
        return dp[current][has] = max (solve (current+1, 1) - v[current] - c, solve (current+1, 0));
}

main () {

    read ();
    printf ("%lld\n", solve (0, 0));
}

```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef pair<ll, ll> ii;
typedef pair<double, double> dd;
typedef vector<ll> vi;
typedef vector<ii> vii;
int dr[] = { 0, 1, -1, 0, 1, -1, -1,  1};
int dc[] = {-1, 0,  0, 1, 1,  1, -1, -1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll INF = 1e14 + 10;
const ll N = 1e2 + 10;
ll color[N], cost[N][N], dp[N][N][N];
ll n, m, k;

void read () {

    scanf ("%lld %lld %lld", &n, &m, &k);

    for (ll i = 0; i < n; i++)
        scanf ("%lld", &color[i]);

    for (ll i = 0; i < n; i++)
        for (ll j = 1; j <= m; j++)
            scanf ("%lld", &cost[i][j]);

    memset (dp, -1, sizeof dp);
}

ll solve (ll current, ll before, ll beauty) {

    if (beauty > k)
        return INF;

    if (current == n)
        return (beauty == k) ? 0LL : INF;

    if (dp[current][before][beauty] != -1)
        return dp[current][before][beauty];

    ll ans = INF;

    if (color[current] == 0)
        for (ll i = 1; i <= m; i++)
            ans = min (ans, cost[current][i] + solve (current+1, i, (i == before) ? beauty : beauty+1));

    else if (current == 0)
        ans = solve (current+1, color[current], 1);

    else
        ans = solve (current+1, color[current], (color[current] == before) ? beauty : beauty+1);

    return dp[current][before][beauty] = ans;
}

int main () {

    read ();
    ll ans = solve (0, color[0], 0);
    printf ("%lld\n", (ans == INF) ? -1 : ans);
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
const int N = 3 * 1e3 + 10;

int r, c, k;

char v[N][N];
int m[N][N];

void read () {

    for (int i = 0; i < r; ++i) {

        getchar ();
        for (int j = 0; j < c; ++j)
            scanf ("%c", &v[i][j]);
    }
}

void make () {

    memset (m, 0, sizeof m);

    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
            m[i+1][j+1] = (v[i][j] == '.' ? 1 : 0) + m[i][j+1] + m[i+1][j] - m[i][j];
}

void solve () {

    int ans = r*c;

    for (int c1 = 0; c1 < c; ++c1) {

        for (int c2 = c1; c2 < c; ++c2) {

            for (int r1 = 0,r2 = 0; r2 < r; ++r2) {

                while (r1 < r2 && m[r2 + 1][c2 + 1] - m[r1 + 1][c2 + 1] - m[r2 + 1][c1] + m[r1 + 1][c1] >= k)
                    ++r1;

                if (m[r2 + 1][c2 + 1] - m[r1][c2 + 1] - m[r2 + 1][c1] + m[r1][c1] >= k)
                    ans = min (ans, (c2 - c1 + 1) * (r2 - r1 + 1));
            }
        }
    }

    printf ("%d\n", ans);
}

main () {

    while (scanf ("%d %d %d", &r, &c, &k)) {

        if (!r && !c && !k)
            break;

        read ();
        make ();
        solve ();
    }
}
```

```cpp
#include <bits/stdc++.h>
#define inf 0x3F3F3F3F
#define mp make pair
using namespace std;

int n, m, k;
char s[111][111];
int id[111][111];
int x[111], y[111];
int dist[20][20];
int vis[111][111];

int di[] = {-2, -2, -1, 1, 2, 2, 1, -1};
int dj[] = {1, -1, 2, 2, 1, -1, -2, -2};

void bfs(int ini) {

    memset (dist[ini], inf, sizeof dist[ini]);
    dist[ini][ini] = 0;

    memset (vis, inf, sizeof vis);
    vis[x[ini]][y[ini]] = 0;

    queue<pair<int, int> > q;
    q.push (mp(x[ini], y[ini]));

    while (!q.empty ()) {

        int i = q.front().first, j = q.front().second;
        q.pop();

        for(int k = 0; k < 8; ++k) {

            int ii = i+di[k], jj = j+dj[k];

            if (ii < 0 || jj < 0 || ii >= n || jj >= m)
                continue;

            if(s[ii][jj] == '#')
                continue;

            if(vis[ii][jj] != inf)
                continue;

            vis[ii][jj] = vis[i][j]+1;
            q.push(mp(ii, jj));

            if (id[ii][jj] != -1)
                dist[ini][ id[ii][jj] ] = vis[ii][jj];
        }
    }
}

int dp[1<<15][16];
int solve (int mask, int pos) {

    if (mask == (1<<k)-1)
        return dist[pos][k];

    int &p = dp[mask][pos];

    if (p != -1) return p;
    p = inf;

    for(int i = 0; i < k; ++i) {

        if (mask & (1<<i))
            continue;

        p = min (p, solve (mask|(1<<i), i) + dist[pos][i]);
    }

    return p;
}

main () {

    while (1) {

        int cnt = 0;
        scanf ("%d %d %d", &n, &m, &k);
        if (!n && !m && !k)
            break;

        memset (id, -1, sizeof id);
        for (int i = 0; i < n; ++i) {

            scanf("%s", s[i]);
            for (int j = 0; j < m; ++j) {

                if (s[i][j] == 'P') {

                    x[cnt] = i;
                    y[cnt] = j;
                    id[i][j] = cnt++;
                }

                if (s[i][j] == 'C') {

                    x[k] = i;
                    y[k] = j;
                    id[i][j] = k;
                }
            }
        }

        for (int i = 0; i <= k; ++i)
            bfs(i);

        memset (dp, -1, sizeof dp);
        printf ("%d\n", solve (0, k));
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
const int alph=256;
const int INF=1e9;
const int MPOW=16;
const int N=1<<MPOW-1;
const int N2=N<<1;
struct sg_tree {

    int arr[N2];

    void build(vector<int> x,int n) {

        fill(arr,arr+N2,INF);

        for(int i=0;i<n;i++)
            arr[i+N] = x[i];

        for(int i=N-1;i>0;i--)
            arr[i] = min(arr[i<<1],arr[(i<<1)+1]);
    }

    int get_min(int c,int cl,int cr,int l,int r) {

        if(l==cl && r==cr)
            return arr[c];

        if(l > r)
            return INF;

        int cm = cl + cr>>1;

        return min(get_min(c<<1,cl,cm,l,min(r,cm)),get_min((c<<1)+1,cm+1,cr,max(l,cm+1),r));
    }

    int get_min (int l,int r) {

        return get_min (1, 0, N-1, l, r);
    }
};

pair<vector<int>,vector<int>> compute(string &s)
{
    int n=s.size();
    int maxn=n+alph;

    vector<int> p(n),c(n),cnt(maxn,0);
    for(int i=0;i<n;i++)
        cnt[s[i]]++;
    for(int i=1;i<maxn;i++)
        cnt[i]+=cnt[i-1];
    for(int i=0;i<n;i++)
        p[--cnt[s[i]]]=i;
    int cl=0;
    c[p[0]]=cl;
    for(int i=1;i<n;i++)
    {
        if(s[p[i]]!=s[p[i-1]])cl++;
        c[p[i]]=cl;
    }
    vector<int> lcp(n,0);
    for(int i=1;i<n;i++)
        lcp[i]=c[p[i]]==c[p[i-1]];
    vector<int> pn(n),cn(n),lcpn(n);
    vector<int> rpos(n),lpos(n);
    sg_tree rmq;
    int k=1;
    while(k<n)
    {
        fill(begin(cnt),end(cnt),0);
        for(int i=0;i<n;i++)
            rpos[c[p[i]]]=i;
        for(int i=n-1;i>=0;i--)
            lpos[c[p[i]]]=i;
        for(int i=0;i<n;i++)
        {
            pn[i]=p[i]-k;
            if(pn[i]<0)pn[i]+=n;
        }
        for(int i=0;i<n;i++)
            cnt[c[i]]++;
        for(int i=1;i<maxn;i++)
            cnt[i]+=cnt[i-1];
        for(int i=n-1;i>=0;i--)
            p[--cnt[c[pn[i]]]]=pn[i];
        cl=0;
        cn[p[0]]=0;
        for(int i=1;i<n;i++)
        {
            int m1=(p[i]+k)%n,m2=(p[i-1]+k)%n;
            if(c[p[i]]!=c[p[i-1]] || c[m1]!=c[m2])cl++;
            cn[p[i]]=cl;
        }
        rmq.build(lcp,n);
        for(int i=1;i<n;i++)
        {
            int a=p[i],b=p[i-1];
            if(c[a]!=c[b])
                lcpn[i]=lcp[lpos[c[a]]];
            else
            {
                int aa=(a+k)%n,bb=(b+k)%n;
                if(c[aa]==c[bb])
                    lcpn[i]=k<<1;
                else
                    lcpn[i]=k+rmq.get_min(lpos[c[bb]]+1,rpos[c[aa]]);
            }
            lcpn[i]=min(n,lcpn[i]);
        }
        copy(begin(cn),end(cn),begin(c));
        copy(begin(lcpn),end(lcpn),begin(lcp));
        k<<=1;
    }
    return {p,lcp};
}


struct suffix_tree
{
    struct edge
    {
        int from;
        int to;
        int next_vert;
        int suffix_here;
    };

    struct vertex
    {
        vector<edge> go;
    };

    string str;
    vector<vertex> data;

    static bool comp(const edge &b,const char &a)
    {
        return 1;
    }

    void build(string &s)
    {
        pair<vector<int>,vector<int>> info=compute(s);
        vector<int> p=info.first,lcp=info.second;

        int n=s.size();
        str=s;

        vector<int> p vert;
        vector<int> p edge;
        vector<int> p dist;


        vertex v;
        edge e;
        e.from=p[0];
        e.to=n;
        e.next vert=-1;
        e.suffix here=p[0];
        v.go.push back(e);
        data.push back(v);

        p vert.push back(0);
        p edge.push back(0);
        p dist.push back(0);

        for(int i=1;i<n;i++)
        {
            int c lcp=lcp[i];

            while(p dist.back()>c lcp)
            {
                edge &E=data[p vert.back()].go[p edge.back()];
                if(E.next vert+1)
                {
                    int m=data[E.next vert].go.size();
                    for(int j=0;j<m;j++)
                        E.suffix here=min(E.suffix here,data[E.next vert].go[j].suffix here);
                }
                p vert.pop back();
                p edge.pop back();
                p dist.pop back();
            }
            vertex v;
            edge e;
            int c v=p vert.back();
            int c e=p edge.back();
            int At=data[c v].go[c e].from+c lcp-p dist.back();

            p dist.push back(c lcp);

            e.next vert=-1;
            e.suffix here=p[i];
            e.from=p[i]+c lcp;
            e.to=n;

            if(At==data[c v].go[c e].from)
            {
                data[c v].go.push back(e);
                p vert.push back(c v);
                p edge.push back(data[c v].go.size()-1);
            }
            else
            {
                v.go.push back(data[c v].go[c e]);
                v.go.back().from=At;
                v.go.push back(e);
                data.push back(v);
                data[c v].go[c e].next vert=data.size()-1;
                data[c_v].go[c_e].to=At;
                p vert.push back(data.size()-1);
                p edge.push back(1);
            }
        }

        while(!p dist.empty())
        {
            edge &E=data[p vert.back()].go[p edge.back()];
            if(E.next vert+1)
            {
                int m=data[E.next vert].go.size();
                for(int j=0;j<m;j++)
                    E.suffix here=min(E.suffix here,data[E.next vert].go[j].suffix here);
            }

            p vert.pop back();
            p edge.pop back();
            p dist.pop back();
        }
    }

    int search str(string &s)
    {
        int n=s.size();
        int cur v=0;
        int cur e;
        char t;

        for(int i=0;i<n;)
        {
            if(cur v==-1)break;
            t=s[i];
            int cur e;
            for(cur e=0;cur e<data[cur v].go.size();cur e++)
                if(str[data[cur v].go[cur e].from]>=t)break;
            for(int j=data[cur v].go[cur e].from;i<n && j<data[cur v].go[cur e].to;j++,i++)
                if(str[j]!=s[i])
                    i=n+1;
            if(i==n)
                return data[cur v].go[cur e].suffix here;
            cur v=data[cur v].go[cur e].next vert;
        }
        return -1;
    }

    int print(int x)
    {
        if(x==-1)return 0;
        int ans=0;
        for(int i=0;i<data[x].go.size();i++)
            ans+=data[x].go[i].to-data[x].go[i].from+print(data[x].go[i].next vert)-(data[x].go[i].to==str.size());
        return ans;
    }
};

int main() {

    ios::sync with stdio(0);
    cin.tie(0);
    string a;
    cin>>a;
    a+='#';
    suffix tree sf;
    sf.build(a);
    cout<<sf.print(0)<<endl;
}
```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 85;
char x[N], y[N];
ll len x, len y;
ll dp[N][N];
ll c = 0;

void print solution () {

    // 1: insert, 2: delete, 3: replace
    stack<ii> s; // first: command second: position

    ll i = len x, j = len y;

    while (true) {

        if (!i and !j)
            break;

        else if (!i)
            s.push (make pair (1LL, i)), --j; // insert

        else if (!j)
            s.push (make pair (2LL, i)), --i; // delete

        else {

            if (dp[i][j] == dp[i-1][j-1] + (x[i-1] != y[j-1])) {

                if (x[i-1] != y[j-1])
                    s.push (make pair (3LL, i-1)); // replace

                --i, --j;
            }

            else if (dp[i][j] - 1 == dp[i][j-1])
                s.push (make pair (1LL, i)), --j; // insert

            else
                s.push (make pair (2LL, i)), --i; // delete
        }
    }

    ll idx = 1;
    ll current = 0;
    while (!s.empty ()) {

        i = s.top ().F; j = s.top ().S;
        s.pop ();

        if (i == 1)
            printf ("%lld Insert %lld,%c\n", idx++, j + current + 1, y[j + current]), ++current;

        if (i == 2)
            printf ("%lld Delete %lld\n", idx++, j + current), --current; // %c, x[j - 1];

        if (i == 3)
            printf ("%lld Replace %lld,%c\n", idx++, j + current + 1, y[j + current]);
    }
}

void solve () {

    len x = strlen (x), len y = strlen (y);

    for (ll i = 0; i <= len x; ++i) {

        for (ll j = 0; j <= len y; ++j) {

            if (!i)
                dp[i][j] = j; // empty (x) -> insert

            else if (!j)
                dp[i][j] = i; // empty (y) -> delete

            else
                dp[i][j] = min (dp[i-1][j-1] + (x[i-1] != y[j-1]), min (dp[i][j-1], dp[i-1][j]) + 1);
                // horizontal (j-1) -----> insert
                // vertical (i-1) -------> delete
                // diagonal (i-1, j-1) --> replace or nothing
        }
    }

    if (c++)
        printf ("\n");

    printf ("%lld\n", dp[len x][len y]);
    print solution ();
}

main () {

    while (gets (x) != NULL) {

        gets (y);
        solve ();
    }
}
```

string - edit distance with transposition.cpp

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e3 + 10;
ll dp[N][N];
string x, y;
ll len x, len y;

void solve () {

    for (ll i = 0; i <= x.length (); ++i) {

        for (ll j = 0; j <= y.length (); ++j) {

            if (!i)
                dp[i][j] = j;

            else if (!j)
                dp[i][j] = i;

            else
                dp[i][j] = min (dp[i-1][j-1] + (x[i-1] != y[j-1]), min (dp[i][j-1], dp[i-1][j]) + 1);

            if (i > 1 and j > 1 and x[i-1] == y[j-2] and x[i-2] == y[j-1])
                dp[i][j] = min (dp[i][j], dp[i-2][j-2] + 1); // transposition
        }
    }

    printf ("%lld\n", dp[x.length ()][y.length ()]);
}

main () {

    while (cin >> x) {

        cin >> y;
        solve ();
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;

ll hamming distance (const char *x, const char *y) {

    ll distance = 0;

    for (ll i = 0; i < strlen (x); ++i)
        if (x[i] != y[i])
            distance++;

    return distance;
}

const ll N = 1e5 + 10;

main() {

    char x[N], y[N]; scanf ("%s", x);
    ll n; scanf ("%lld", &n);
    ll position = 1;
    ll distance = N;

    for (ll i = 1; i <= n; ++i) {

        scanf ("%s", y);
        ll d = hamming distance (x, y);

        if (d < distance) {

            position = i;
            distance = d;
        }
    }

    printf ("%lld\n", position);
    printf ("%lld\n", distance);
}

```

string - hashing.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
typedef long long unsigned int llu;
const int N = 1e5 + 10;

llu hashString (char *str) { // base 33

    llu hash = 5381;
    int c;

    while (c = *str++)
        hash = ((hash << 5) + hash) + c;

    return hash;
}

main () {

    map<llu, llu> okay;
    okay.clear ();

    int n; scanf ("%d", &n);
    while (n--) {

        char s[N]; scanf ("%s", s);

        llu hash = hashString (s);
        if (okay.find (hash) == okay.end ()) {

            printf ("OK\n");
            okay[hash] = 1;
        }

        else {

            printf ("%s%llu\n", s, okay[hash]);
            okay[hash]++;
        }
    }
}

```

```
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 100;
ll back table[N];
ll len x, len y, ans;
string x, y;

void make () {

    ll i = 0, j = -1;
    back table[0] = -1;

    while (i < len y) {

        while (j >= 0 and y[i] != y[j])
            j = back table[j];

        i++; j++;
        back table[i] = j;
    }
}

void kmp () {

    ans = 1;
    ll i = len y, j = 0; // start after end of len x
    while (i < len x) {

        while (j >= 0 and x[i] != y[j])
            j = back table[j];

        i++; j++;

        if (j == len y) {

            ans++;
            j = back table[j];
        }
    }
}

main () {

    ll c = 0;
    ll t; scanf ("%lld", &t);
    while (t--) {

        cin >> x;
        len x = x.length ();
        len y = 0;

        while (true) {

            ++len y;
            if (len x % len y)
                continue;

            y = x.substr (0, len y);
            make ();
            kmp ();

            if (ans == len x / len y)
                break;
        }

        if (c++)
            printf ("\n");
        printf ("%lld\n", len y);
    }
}

```

string - lcs with at at least k length segments.cpp

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e3 + 10;
ll n;
char x[N], y[N];
ll len x, len y;
ll dp[N][N];
void solve ();

main () {

    while (scanf ("%lld", &n), n)
        solve ();
}

void solve () {

    scanf ("%s %s", x, y);
    len x = strlen (x), len y = strlen (y);

    for (ll i = 1; i <= len x; ++i) {

        for (ll j = 1; j <= len y; ++j) {

            ll ans = 0;
            dp[i][j] = max (dp[i-1][j], dp[i][j-1]);

            while (i-1-ans >= 0 and j-1-ans >= 0 and x[i-1-ans] == y[j-1-ans]) {

                ++ans;
                if (ans >= n)
                    dp[i][j] = max (dp[i][j], dp[i-ans][j-ans] + ans);
            }

            if (ans >= n)
                dp[i][j] = max (dp[i][j], dp[i-ans][j-ans] + ans);
        }
    }

    printf ("%lld\n", dp[len x][len y]);
}
```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define eoq cout << "eoq" << endl
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 3 * 1e3 + 10;
ll c = 1;
ll dp[N][N];
map<llu, string> word;

llu encrypt string (const char *str) { // base 33

    llu hash = 5381;
    llu c;

    while (c = *str++)
        hash = ((hash << 5) + hash) + c;

    return hash;
}

vector<llu> read () {

    string s;
    vector<llu> v;
    while (cin >> s and s != "#") {

        llu hash = encrypt string (s.c str ());
        word[hash] = s;
        v.push back (hash);
    }

    return v;
}

void print dp (vector<llu> x, vector<llu> y, ll i, ll j) {

    stack<string> s;
    while (i and j) {

        if (dp[i][j] == dp[i-1][j])
            i--;

        else if (dp[i][j] == dp[i][j-1])
            j--;

        else {

            s.push (word[x[i-1]]);
            i--;
            j--;
        }
    }

    if (!s.empty ()) {

        cout << s.top ();
        s.pop ();
    }

    while (!s.empty ()) {

        cout << ' ' << s.top ();
        s.pop ();
    }

    cout << '\n';
}

void solve (vector<llu> x, vector<llu> y) {

    for (ll i = 0; i <= x.size (); ++i) {

        for (ll j = 0; j <= y.size (); ++j) {

            if (!i || !j)
                dp[i][j] = 0;

            else if (x[i-1] == y[j-1])
                dp[i][j] = dp[i-1][j-1] + 1;

            else
                dp[i][j] = max (dp[i-1][j], dp[i][j-1]);
        }
    }

    print dp (x, y, x.size (), y.size ());
}

main () {

    ios base::sync with stdio (0);
    cin.tie (0);
    cin.tie (0);

    while (true) {

        word.clear ();
        vector<llu> x = read ();
        if (x.size () == 0)
            break;

        vector<llu> y = read ();
        solve (x, y);
    }
}

```

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define mp make_pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 250*250 + 10;
ll idx[N], dp[N], tail[N];
ll n, p, q;
ll c = 1;

void read () {

    memset (idx, -1, sizeof idx);
    scanf ("%lld %lld %lld", &n, &p, &q);
    for (ll i = 0; i <= p; ++i) {

        ll x; scanf ("%lld", &x);
        idx[x] = i;
    }

    for (ll i = 0; i <= q; ++i) {
        // vector of positions where elem 'x' were found
        ll x; scanf ("%lld", &x);
        dp[i] = idx[x];
    }
}

ll get_index (int lo, int hi, int key) {

    while (hi-lo > 1) {

        ll mid = lo + ((hi-lo) >> 1);

        if (tail[mid] >= key)
            hi = mid;

        else
            lo = mid;
    }

    return hi;
}

void solve () {

    // n log n LIS on modified vector
    memset (tail, 0, sizeof tail);
    ll length = 1;
    tail[0] = dp[0];
    for (ll i = 1; i <= q; i++) {

        if (dp[i] < tail[0])
            tail[0] = dp[i];

        else if (dp[i] > tail[length-1])
            tail[length++] = dp[i];

        else
            tail[get_index (-1, length-1, dp[i])] = dp[i];
    }

    printf ("Case %lld: %lld\n", c++, length);
}

main () {

    ll t; scanf ("%lld", &t);
    while (t--) {

        read ();
        solve ();
    }
}

```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 1e3 + 10;
ll c = 1;
ll dp[N][N];

void solve (string x, string y) {

    ll len x = x.length ();
    ll len y = y.length ();

    for (ll i = 0; i <= len x; ++i) {

        for (ll j = 0; j <= len y; ++j) {

            if (!i || !j)
                dp[i][j] = 0;

            else if (x[i-1] == y[j-1])
                dp[i][j] = dp[i-1][j-1] + 1;

            else
                dp[i][j] = max (dp[i-1][j], dp[i][j-1]);
        }
    }

    printf ("Case #%lld: you can visit at most %lld cities.\n", c++,
    dp[len x][len y]);
}

main () {

    string x, y;
    while (getline (cin, x), x != "#") {

        getline (cin, y);
        solve (x, y);
    }
}

```

```cpp
#include <bits/stdc++.h>
#define pb push back
#define mp make pair
#define F first
#define S second
#define oioi printf("oioi\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef vector<ll> vi;
typedef pair<ll, ll> ii;
typedef vector< ii > vii;
int dr[] = {0 ,1,-1,0,1,-1,-1, 1};
int dc[] = {-1,0,0, 1,1, 1,-1,-1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ll N = 2 * 1e3 + 10;
char special[N], x[N]; ll len;
ii dp[N][N]; // F-> special positions, S -> total palindrome

void read () {

    scanf ("%s", x);
    len = strlen (x);

    memset (special, false, sizeof special);
    ll n; scanf ("%lld", &n);
    for (ll i = 0; i < n; ++i) {

        ll x; scanf ("%lld", &x);
        special[x-1] = true;
    }
}

void solve () {

    for (ll k = 1; k <= len; ++k) {

        for (ll i = 0; i < len - k + 1; ++i) {

            if (k == 1) {

                dp[i][i].F = special[i];
                dp[i][i].S = 1;
                continue;
            }

            ii a; a.F = a.S = 0;
            ll j = i + k - 1;

            if (x[i] == x[j]) {

                if (k > 2)
                    a = dp[i+1][j-1];

                a.F += special[i] + special[j];
                a.S += 2;
            }

            dp[i][j] = max (a, max (dp[i+1][j], dp[i][j-1]));
        }
    }

    printf ("%lld\n", dp[0][len - 1].S);
}

main () {

    read ();
    solve ();
}

```

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define mp make_pair
#define F first
#define S second
#define eoq printf("eoq\n")
#define digitCountDec(a) (int)floor(1 + log10((double)a))
#define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
using namespace std;
typedef unsigned long long int llu;
typedef long long int ll;
typedef pair<ll, ll> ii;
typedef pair<double, double> dd;
typedef vector<ll> vi;
typedef vector<ii> vii;
int dr[] = { 0, 1, -1, 0, 1, -1, -1,  1};
int dc[] = {-1, 0,  0, 1, 1,  1, -1, -1};

ll gcd (ll x, ll y) {

    return (y != 0 ? gcd (y, x%y) : x);
}

ll lcm (ll x, ll y) {

    return (x / gcd (x, y) * y);
}

const ii zero = mp (0, 0);
const ll N = 2 * 1e3 + 10;
char s[N], special[N];
ll len;
ii dp[N][N];

void read () {

    memset (special, false, sizeof special);

    ll n; scanf ("%s %lld", s, &n);
    while (n--) {

        ll x; scanf ("%lld", &x);
        special[--x] = true;
    }

    len = strlen (s);
    memset (dp, -1, sizeof dp);
}

ii solve (ll i, ll j) {

    if (i > j)
        return dp[i][j] = zero;

    if (i == j)
        return dp[i][j] = mp (special[i], 1);

    if (dp[i][j].F != -1)
        return dp[i][j];

    dp[i][j] = max (solve (i+1, j), solve (i, j-1));

    if (s[i] == s[j]) {

        ii ans = solve (i+1, j-1);
        return dp[i][j] = max (dp[i][j], mp (ans.F + special[i] + special[j], ans.S + 2));
    }

    return dp[i][j];
}
```

```cpp

main () {

    read ();
    printf ("%lld\n", solve (0, len-1).S);
}

```

```cpp
#include <bits/stdc++.h>
#define F first
#define S second
using namespace std;
typedef long long int ll;

const ll N = 1e3 + 10;
int caso = 1;
char v[N][N];
string s1, s2;
pair<int, int> dp[N][N];

void solve () {

    int n = s1.length ();
    int m = s2.length ();

    for (int i = n; i >= 0; --i) {

        for (int j = m; j >= 0; --j) {

            if (i == n and j == m) {

                dp[i][j].F = 0;
                dp[i][j].S = 0;
            }

            else if (i == n) {

                dp[i][j].F = 0;
                dp[i][j].S = m-j;
            }

            else if (j == m) {

                dp[i][j].F = n-i;
                dp[i][j].S = 0;
            }

            else if (s1[i] == s2[j]) {

                dp[i][j].F = dp[i+1][j+1].F;
                dp[i][j].S = dp[i+1][j+1].S;
            }

            else if (dp[i+1][j].F + dp[i+1][j].S <= dp[i][j+1].F + dp[i][j+1].S) {

                dp[i][j].F = dp[i+1][j].F + 1;
                dp[i][j].S = dp[i+1][j].S;
            }

            else {

                dp[i][j].F = dp[i][j+1].F;
                dp[i][j].S = dp[i][j+1].S + 1;
            }
        }
    }

    printf ("Case %d: %d %d\n", caso++, dp[0][0].F , dp[0][0].S);
}

main () {

    int t; scanf ("%d", &t);
    while (t--) {

        int h, w; scanf ("%d %d", &h, &w);

        for (int i = 0; i < h; ++i) {

            getchar ();
            for (int j = 0; j < w; ++j)
                scanf ("%c", &v[i][j]);
        }

        int n, x, y;
        scanf ("%d %d %d", &n, &x, &y); getchar ();
        --x; --y;
        s1.clear ();
        s1.push back (v[y][x]);
        while (n--) {

            char c; scanf ("%c", &c);
            switch (c) {

                case 'N':
                    --y;
                    break;

                case 'E':
                    ++x;
                    break;

                case 'W':
                    --x;
                    break;

                case 'S':
                    ++y;
                    break;
            }

            s1.push back (v[y][x]);
        }

        scanf ("%d %d %d", &n, &x, &y); getchar ();
        --x; --y;
        s2.clear ();
        s2.push back (v[y][x]);
        while (n--) {

            char c; scanf ("%c", &c);
            switch (c) {

                case 'N':
                    --y;
                    break;

                case 'E':
                    ++x;
                    break;

                case 'W':
                    --x;
                    break;

                case 'S':
                    ++y;
                    break;
            }

            s2.push back (v[y][x]);
        }

        solve ();
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

int n;
string s;
int suffixArr[100010];

struct suffix {

    int index;
    int rank[2];
};

int compare (struct suffix a, struct suffix b) {

    return (a.rank[0] == b.rank[0]) ? (a.rank[1] < b.rank[1] ? 1 : 0) : (a.rank[0] < b.rank[0] ? 1 : 0);
}

void build () {

    n = s.length ();
    struct suffix suffixes[n];

    for (int i = 0; i < n; ++i) {

        suffixes[i].index = i;
        suffixes[i].rank[0] = s[i] - 'a';
        suffixes[i].rank[1] = ((i+1) < n)? (s[i + 1] - 'a'): -1;
    }

    sort (suffixes, suffixes+n, compare);

    int ind[n];

    for (int k = 4; k < 2*n; k = k*2) {

        int rank = 0;
        int prev rank = suffixes[0].rank[0];
        suffixes[0].rank[0] = rank;
        ind[suffixes[0].index] = 0;

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

            if (suffixes[i].rank[0] == prev rank && suffixes[i].rank[1] == suffixes[i-1].rank[1]) {

                prev rank = suffixes[i].rank[0];
                suffixes[i].rank[0] = rank;
            }

            else {

                prev rank = suffixes[i].rank[0];
                suffixes[i].rank[0] = ++rank;
            }

            ind[suffixes[i].index] = i;
        }

        for (int i = 0; i < n; ++i) {

            int nextindex = suffixes[i].index + k/2;
            suffixes[i].rank[1] = (nextindex < n) ? suffixes[ind[nextindex]].rank[0] : -1;
        }

        sort (suffixes, suffixes+n, compare);
    }

    for (int i = 0; i < n; ++i)
```

```cpp
        suffixArr[i] = suffixes[i].index;
}

void solve () {

    vector<int> lcp (n, 0);
    vector<int> invSuff (n, 0);

    for (int i = 0; i < n; ++i)
        invSuff[suffixArr[i]] = i;

    int k = 0;
    int ans = 0, pos = 0;

    for (int i = 0; i < n; ++i) {

        if (invSuff[i] == n-1) {

            k = 0;
            continue;
        }

        int j = suffixArr[invSuff[i]+1];
        while (i+k < n and j+k < n and s[i+k] == s[j+k])
            k++;

        lcp[invSuff[i]] = k;

        if (k > ans)
            pos = i, ans = k;

        if (k > 0)
            k--;
    }

    if (ans > 2)
        cout << s.substr (pos, ans) << '\n';

    else
        cout << '*' << '\n';
}

main () {

    cin >> s;
    build ();
    solve ();
}
```