A.

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

using namespace std;
const int N = int(2e5) + 10, mod = int(1e9) + 7;

int n, q, a[N], was[N];

vector <int> d[N];

int main(){
    int T = 1;
    cin.tie(NULL);
    cout.tie(NULL);
    ios_base::sync_with_stdio(false);

    cin >> n >> q;

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

    for(int i = 2; i < N; i++) if(d[i].empty()){
        for(int j = i; j < N; j += i){
            d[j].push_back(i);
        }
    }

    for(int it = 1, x; it <= q; it++){
        cin >> x;
        bool f = 0;
        for(int i = 0; i < n && !f; i++){
            for(int p : d[a[i] + x]){
                if(was[p] == it){
                    f = 1;
                    break;
                }
                was[p] = it;
            }
        }
        if(f) cout << "NO\n";
        else cout << "YES\n";
    }


    return 0;
}
```

B.

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

using namespace std;

const int N = int(1e6) + 1;

int n,k, s[N], p[N];

int main(){
    cin >> n >> k;

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

    for(int i = 1; i < N; i++) p[i] += p[i - 1];

    int id = k;

    long long ans = 0;

    for(int x = k + 1; x < N; x++){
        long long cur = 0;
        for(int i = 1; i * x < N; i++){
            cur += 1ll * i * (p[min(i * x + x - 1, N - 1)] - p[i * x - 1]);
        }
        cur *= (x - k);
        if(cur > ans){
            ans = cur;
            id = x;
        }
    }
    cerr << id << endl;
    cout << ans;


    return 0;
}
```

C.

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

#define ll long long
#define f first
#define s second
#define pb push_back
#define all(x) x.begin(), x.end()

using namespace std;
const int N = int(3e5) + 10, mod = int(1e9) + 7;
const ll inf = (ll)(1e18);

int n, x[N], y[N], pos[N];

pair <int,int> p[N], q[N];

int t[4*N], u[4*N], c[2*N], p2[N];

void add(int &a,int b){
	a += b;
	if(a >= mod) a -= mod;
}

void push(int v){
	if(u[v] == 1) return;
	t[v + v] = 1ll * t[v + v] * u[v] % mod;
	t[v + v + 1] = 1ll * t[v + v + 1] * u[v] % mod;
	u[v + v + 1] = 1ll * u[v + v + 1] * u[v] % mod;
	u[v + v] = 1ll * u[v + v] * u[v] % mod;
	u[v] = 1;
}

void updpos(int v,int tl,int tr,int pos, int val){
	if(tl == tr){
		t[v] = val;
		return;
	}
	int tm = (tl + tr) >> 1;
	push(v);
	if(pos <= tm) updpos(v + v, tl, tm, pos, val);
	else updpos(v + v + 1, tm + 1, tr, pos, val);
	t[v] = (t[v + v] + t[v + v + 1]) % mod;
}

void upd(int v,int tl,int tr,int l,int r){
    if(l > tr || tl > r) return;
    if(l <= tl && tr <= r){
        add(t[v], t[v]);
        add(u[v], u[v]);
        return;
    }
    int tm = (tl + tr) >> 1;
    push(v);
    upd(v + v, tl, tm, l, r);
    upd(v + v + 1, tm + 1, tr, l, r);
    t[v] = (t[v + v] + t[v + v + 1]) % mod;
}

int get(int v,int tl,int tr,int l,int r){
    if(l > tr || tl > r) return 0;
    if(l <= tl && tr <= r) return t[v];
    int tm = (tl + tr) >> 1;
    push(v);
    return (get(v + v, tl, tm, l, r) + get(v + v + 1, tm + 1, tr, l, r)) % mod;
}

void build(int v,int tl,int tr){
    u[v] = 1;
    t[v] = 0;
    if(tl == tr) return;
    int tm = (tl + tr) >> 1;
    build(v + v, tl, tm);
    build(v + v + 1, tm + 1, tr);
}

void upd(int v){
    v += n;
    while(v){
        c[v]++;
        v /= 2;
    }
}

int get(int l,int r){
    l += n;
    r += n;
    int res = 0;
    while(l <= r){
        if(l & 1) res += c[l];
        if(!(r & 1)) res += c[r];
        l = (l + 1) >> 1;
        r = (r - 1) >> 1;
    }
```

```
        return res;
}

long long calc(int nx,int ny){
        for(int i = 0; i < n; i++){
                q[i] = {ny * y[i], i};
                p[i] = {nx * x[i], i};
        }
        sort(q, q + n);
        for(int i = 0; i < n; i++){
                pos[q[i].s] = i;
        }
        sort(p, p + n);
        ll res = 0;
        memset(c, 0, sizeof(c));
        build(1, 0, n - 1);
        for(int i = 0; i < n; i++){
                int val = get(0, pos[p[i].s] - 1);
                upd(pos[p[i].s]);


                val = 1ll * p2[val] * q[pos[p[i].s]].f % mod;
                if(val < 0) val += mod;
                updpos(1, 0, n - 1, pos[p[i].s], val);
                res += 1ll*p[i].f * get(1, 0, n - 1, pos[p[i].s], n - 1) % mod;
                upd(1, 0, n - 1, pos[p[i].s] + 1, n - 1);
                res %= mod;
        }
        return res;
}

void solve(){
        cin >> n;

        p2[0] = 1;
        for(int i = 1; i <= n; i++){
                p2[i] = p2[i - 1];
                add(p2[i], p2[i]);
        }

        for(int i = 0; i < n; i++){
                cin >> x[i] >> y[i];
        }
        //cout << calc(1,1) << endl;
        //return;
        cout << ((calc(1,1) + calc(1, -1) + calc(-1, 1) + calc(-1, -1)) % mod + mod) % mod;
}
```

```
int main(){
	int T = 1;
	cin.tie(NULL);
	cout.tie(NULL);
	ios_base::sync_with_stdio(false);

	//cin >> T;
	while(T--){
		solve();
	}
	return 0;
}
```

D.

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

#define all(x) (x).begin(), (x).end()
#define sz(x) (int)(x).size()

using namespace std;

typedef long long ll;

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    int n, m, a, b;
    cin >> n >> m >> a >> b;
    ll t = 0;
    while (n > 0 || m > 0) {
        if (abs(n - m) <= 1) {
            ll p = (t + a) / a;
            ll q = (t + b) / b;
            t = min(a * p, b * q);
        }
        else {
            t++;
        }

        if (t % a == 0 && n > 0) {
            n--;
        }

        if (t % b == 0 && m > 0) {
            m--;
        }

        if (n > m + 1) {
            n--;
            m++;
        }
        else if (m > n + 1) {
            m--;
            n++;
        }
    }
    cout << t << '\n';
    return 0;
}
```

E.

```
#include <bits/stdc++.h>
using namespace std;
const int N = int(1e5) + 10;
int n, m, b, used[N], col[N];
vector <int> g[N], ord;
void dfs(int v){
	used[v] = 1;
	for(int to : g[v]) if(!used[to]) dfs(to);
	ord.push_back(v);
}
void solve(){
	cin >> n >> m >> b;
	for(int i = 1,u,v; i <= m; i++){
		cin >> u >> v;
		g[u].push_back(v);
	}
	col[1] = 1; b--;
	for(int i = 1; i <= n; i++)
		if(!used[i]) dfs(i);
	reverse(ord.begin(), ord.end());
	for(int v : ord){
		if(!col[v] && b){
			col[v] = 1; b--;
		}
	}
	for(int i = 1; i <= n; i++)
		if(col[i]) cout << i << " ";
	cout << "\n";
	ord.clear();
	for(int i = 1; i <= n; i++) {
		used[i] = 0;
		col[i] = 0;
		g[i].clear();
	}
}
int main(){
	int T = 1;
	cin.tie(NULL);
	cout.tie(NULL);
	ios_base::sync_with_stdio(false);
	cin >> T;
	while(T--){
		solve();
	}
	return 0;
}
```

F.

```
#include <bits/stdc++.h>
using namespace std;
const int N = 500500;
const int LOG = 20;
int cnt[N + 1];
int a[LOG][N];
inline int solve(int l, int r) {
    int len = r - l;
    int res = 0;
    for (int k = LOG - 1; k >= 0; k--) if ((len >> k) & 1) {
        res ^= a[k][l];
        l += 1 << k;
        res ^= (cnt[l] ^ cnt[r]) << k;
    }
    return res;
}
int main()
{
    int n, m, x;
    scanf("%d%d", &n, &m);
    while(n--) {
        scanf("%d", &x);
        cnt[x + 1] ^= 1;
    }
    for (int i = 1; i <= N; i++)
        cnt[i] ^= cnt[i - 1];
    for (int k = 0; k < LOG - 1; k++)
        for (int i = 0; i < N; i++) {
            a[k + 1][i] = a[k][i];
            x = i + (1 << k);
            if (x >= N) continue;
            a[k + 1][i] ^= a[k][x];
            a[k + 1][i] ^= (cnt[x] ^ cnt[min(x + (1 << k), N)]) << k;
        }
    for (int k = 1; k <= m; k++) {
        int ans = 0;
        int l = 0;
        while(l < N) {
            ans ^= solve(l, min(N, l + k));
            l += k;
        }
        printf("%d ", ans);
    }
    printf("\n");
    return 0;
}
```