

# D

```
//Solution by Tima
#include <bits/stdc++.h>

#define f first
#define s second
#define ll long long
#define ull unsigned long long
#define mp make_pair
#define pb push_back
#define vi vector<int>
#define ld long double
#define pii pair<int, int>
#define y1 sda
#define all(x) x.begin(), x.end()

using namespace std;
const int N = 512, mod = int(1e9) + 7;

int n,m,k;

int d[N][N], a[N][N], b[N][N];

int calc(int x){
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            b[i][j] = max(0, a[i][j] - x);
        }
    }

    for(int i = 0; i <= n; i++){
        for(int j = 0; j <= m; j++) d[i][j] = mod;
    }

    d[0][1] = 0;

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            d[i][j] = min(d[i - 1][j], d[i][j - 1]) + b[i][j];
        }
    }
    return d[n][m] + k * x;
}

int main () {
    scanf("%d%d%d", &n, &m, &k);
```

```

        for(int i = 1; i <= n; i++)
            for(int j = 1; j <= m; j++)
                scanf("%d", &a[i][j]);

vector <int> distinct;

for(int i = 1; i <= n; i++)
    for(int j = 1; j <= m; j++)
        distinct.pb(a[i][j]);

sort(all(distinct));
distinct.resize(unique(all(distinct)) - distinct.begin());

int res = mod;
for(int x : distinct) res = min(res, calc(x));
printf("%d", res);
return 0;
}

```

## E

```

#include <bits/stdc++.h>

#define pb push_back
#define mp make_pair
#define all(x) (x).begin(), (x).end()
#define sz(x) (int)(x).size()

using namespace std;

typedef long long ll;

const int MAXN = (int)3e5 + 5;

int a[MAXN];
int b[MAXN];
int n;

pair<int, int> t[MAXN * 2];
int e[MAXN * 2];

void build(int v = 1, int tl = 1, int tr = n) {
    t[v] = {0, tr - tl + 1};
    e[v] = 0;

    if (tl != tr) {

```

```

int mid = (tl + tr) >> 1;
int c1 = v + 1, c2 = v + (mid + 1 - tl) * 2;

build(c1, tl, mid);
build(c2, mid + 1, tr);
}
}

void push(int v, int tl, int tr) {
    if (e[v]) {
        t[v].first += e[v];

        if (tl != tr) {
            int mid = (tl + tr) >> 1;
            int c1 = v + 1, c2 = v + (mid + 1 - tl) * 2;

            e[c1] += e[v];
            e[c2] += e[v];
        }

        e[v] = 0;
    }
}

pair<int, int> comb(pair<int, int> a, pair<int, int> b) {
    if (a.first == b.first) {
        return mp(a.first, a.second + b.second);
    }

    return max(a, b);
}

void update(int l, int r, int x, int v = 1, int tl = 1, int tr = n) {
    push(v, tl, tr);

    if (l > r || tl > r || tr < l) {
        return;
    }

    if (l <= tl && tr <= r) {
        e[v] += x;
        push(v, tl, tr);
        return;
    }

    int mid = (tl + tr) >> 1;
    int c1 = v + 1, c2 = v + (mid + 1 - tl) * 2;

```

```

update(l, r, x, c1, tl, mid);
update(l, r, x, c2, mid + 1, tr);

t[v] = comb(t[c1], t[c2]);
}

int query() {
    push(1, 1, n);
    return t[1].first == 2 ? t[1].second : 0;
}

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    cin >> n;

    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        b[i] = i;
    }

    sort(b + 1, b + n + 1, [&](int i, int j) -> bool {
        return mp(a[i], i) > mp(a[j], j);
    });

    vector<array<int, 5>> rectangles;

    for (int step = 0; step < 2; step++) {
        set<int> S{0, n+1};

        for (int i = 1; i <= n; i++) {
            int pos = b[i];
            auto it = S.insert(pos).first;

            if (*prev(it) > 0) {
                int lx = *prev(it, 2) + 1;
                int rx = *prev(it);

                int ly = *it;
                int ry = *next(it) - 1;

                int val = abs(a[*prev(it)] - a[pos]);

                rectangles.pb({val, lx, rx, ly, ry});
            }

            if (*next(it) <= n) {

```

```

    int lx = *prev(it) + 1;
    int rx = *it;

    int ly = *next(it);
    int ry = *next(it, 2) - 1;

    int val = abs(a[*next(it)] - a[pos]);

    rectangles.pb({val, lx, rx, ly, ry});
}
}

reverse(b + 1, b + n + 1);
}

ll ans = 0;

build();
sort(all(rectangles));
vector<array<int, 4>> ev;

for (int l = 0, r; l < sz(rectangles); l = r) {
    ev.clear();
    r = l;

    while (r < sz(rectangles) && rectangles[l][0] == rectangles[r][0]) {
        int lx = rectangles[r][1], rx = rectangles[r][2];
        int ly = rectangles[r][3], ry = rectangles[r][4];

        ev.pb({lx, 1, ly, ry});
        ev.pb({rx + 1, -1, ly, ry});

        r++;
    }

    sort(all(ev));
    int lst = -1;

    for (auto [x, val, l, r] : ev) {
        if (lst != -1 && x != lst) {
            ans += query() * 1ll * (x - lst);
        }

        lst = x;
        update(l, r, val);
    }
}

```

```

    cout << ans << '\n';
    cerr << (double)clock() / CLOCKS_PER_SEC << endl;
    return 0;
}

```

## F

```

//by paradox & gege & parasat
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
//using namespace __gnu_pbds;

//#pragma GCC optimize("Ofast")
//#pragma comment(linker, "/stack:200000000")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4")

#define file(s) freopen(s".in","r",stdin); freopen(s".out","w",stdout);
#define fastio ios_base::sync_with_stdio(0), cin.tie(0), cout.tie(0);
#define all(x) x.begin(), x.end()
#define sz(s) (int)s.size()
#define pb push_back
#define ppb pop_back
#define mp make_pair
#define se second
#define fi first
#define s second
#define f first

typedef pair < int, int > pii;
typedef vector < int > vi;
typedef long double ld;
typedef long long ll;

//typedef tree < int, null_type, less < int >, rb_tree_tag, tree_order_statistics_node_update >
//ordered_set;

const int dx[] = {1, -1, 0, 0}, dy[] = {0, 0, 1, -1}, block = 300;
const pii base = mp(1e6 + 3, 1e6 + 7), mod = mp(1e9 + 7, 1e9 + 9);
const int N = 2e5, inf = 1e9 + 1, MOD = 1e9 + 7;
const int M = N + 17;
const ll INF = 1e18;

```

```

int n, k;
int c[M];
bool was[M];

pair<ll, int> get(int pen){
    priority_queue<pii> q;
    q.push({c[1] + pen * was[1], was[1]});
    ll res = 0;
    int cnt = 0;

    for(int i = 1; i <= n; i += 2){
        auto val = q.top();
        q.pop();
        res += val.f;
        cnt += val.s;

        if(i + 1 <= n) q.push({c[i + 1] + pen * was[i + 1], was[i + 1]});
        if(i + 2 <= n) q.push({c[i + 2] + pen * was[i + 2], was[i + 2]});
    }
    return {res, cnt};
}

```

```

ll get(pair<ll, int> a, int pen){
    return a.f - pen * 1ll * a.s;
}

```

```

void solve() {
    scanf("%d%d", &n, &k);

    for(int i = 1; i <= n; i++)
        scanf("%d", &c[i]);
    for(int i = 1; i <= n; i++)
        scanf("%d", &was[i]);
}

```

```

int mx = get(-inf).s;
int mn = get(inf).s;

if(k < mn || k > mx){
    printf("-1\n");
    return;
}

```

```

int l = -inf, r = inf;
while(l <= r){
    int mid = (l + r) >> 1;
}

```

```

        if(get(mid).s <= k) l = mid + 1;
        else r = mid - 1;
    }

    int pen = l - 1;
    pair<ll, int> cur = get(pen);
    ll res = 0;

    if(cur.s == k){
        res = cur.f - k * 1ll * pen;
    }else{
        pair<ll, int> nxt = get(pen + 1);

        ll val1 = get(nxt, l), val2 = get(cur, l - 1);
        ll tot = val1 - val2;
        int delta = (nxt.s - cur.s);
        assert(delta != 0);

        ll d = tot / delta;
        res = d * (k - cur.s) + val2;
    }

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

main () {
    int TT;
    TT = 1;
    for (int tt = 1; tt <= TT; ++tt) {
        solve();
    }
}

```