

```

a) N = int(input())
sum = 0
for i in range(1, N+1):
    sum += 0
    for j in range(1, i+1):
        if i % j == 0:
            sum += 1
    sum += 1
    sum += 1
print(sum)

int a[M][N]
int a[M][N]
int main() {
    cin >> n;
    for (int i = 1; i <= n; i++) {
        sort(a[i], a[i+1]);
        if (a[i][n] <= 0) {
            cout << "Yes"
            return 0;
        }
        i++;
    }
}

```

```

while (t <= 1000000000) {
    if (t > 0) {
        ?
        2 [t] [int] i = 2; i <= n; i++) {
            a[i] = a[i] - 1;
        }
        ? values:
        for (int i = 1; i <= n; i++) {
            values: a[i] * 2;
        }
        sort(values);
        set <int> s;
        for (int i = 1; i <= n; i++) {
            s.insert(a[i]);
        }
        ? ans = 0;
        int ptn = 0;
        for (int i = 1; i <= n; i++) {
            while (ptn <= values - s.values) {
                ptn = values - s.values;
                s.erase(i);
            }
        }
        print
    }
}

```

for i in range(1, num/2):

num[i] + num[num - i] = num

res += 1

print

```
a) N = int(input())
```

```
Sum = 0
```

```
For i in range(1, N+1):
```

```
    sum I = 0
```

```
    For j in range(1, i+1):
```

```
        if i % j == 0:
```

```
            sum I += 1
```

```
        if sum I % 2 == 0:
```

```
            sum + 1
```

```
        sum + 1
```

```
print(Sum)
```

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

```
# define pb push_back
```

```
# define mp make_pair
```

```
# define all(v) begin(v), end(v)
```

```
# define sz(x) (int)(x).size()
```

```
using namespace std
```

```
typedef long long ll;
```

```
const int MAXN = (int) 1e6 + 5
```

```
const int MOD = (int) 1e9 + 7
```

```
int a[MAXN]
```

```
int n;
```

```
int main()
```

```
{
```

```
ios::sync_with_stdio(0)
```

```
cin >> n;
```

Парақтың артқы жағын толтырмаңыз / Обратную сторону листа не заполнять

```
int n;
```

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

```
    cin >> a[i];
```

```
}
```

```
sort(a+1, a+n+1);
```

```
if(a[n] <= 0) {
```

```
    cout << "1\n";
```

```
    return 0;
```

```
} int t = n;
```

```
while (t > 2 * a[t+2] + a[t+1] > a[t] * 2)
```

```
{ t-- }
```

```

a) N = int(input())

```

```

sum = 0

```

```

for i in range(1, N+1):

```

```

    sum1 = 0

```

```

    for j in range(1, i+1):

```

```

        if i % j == 0:

```

```

            sum1 += 1

```

```

        if sum1 % 2 == 0:

```

```

            sum += 1

```

```

            sum += 1

```

```

print(sum)

```

```

b) num = [i for i in range(1, int(input()))]

```

```

    res = 1

```

```

    for i in range(1, len(num), 2):

```

```

        num[i], num[res] = num[res], num[i]

```

```

        res += 2

```

```

    if

```

```

print(num)

```

```

c) const int MaxN = (int) 1e5 + 3;

```

```

    const int k = 300;

```

```

    const int l = MaxN / k + 2;

```

```

    int p[l][MaxN], q[l][l];

```

```

    int

```

```

    int s[MaxN], t[MaxN];

```

```

    int n[MaxN];

```

```

    int m, q;

```

$T[p/k] += x;$

void sqrt Upd1(int p, int x) {
 sp[p] += x;

void sqrt Upd(int l, int r, int l1, int r1, int x)
 int cl = l/k, cr = r/k

if (cl == cr)

for (int i = l; i <= r; i++) {
 p[id][i] += x;

return;

for (int i = l, j = (cl+1)*k; i < j; i++)
 p[id][i] += x;

for (int i = cl+1; i <= cr; i++)
 p[id][i] += x

for (int i = cr*k; i <= r; i++)
 p[id][i] += x;

// sqrt & Get(int l, int r)
int cl = l/k, cr = r/k
if (cl == cr)

if (cl == cr)

for (int i = l; i <= r; i++)

```

return s[i];
return res;
for(int i=1, j=(l+1)*k; i<j; i++)
    res += s[i];
for(int i=l+1; i<=r; i++)
    res += t[i];
for(int i=c*k; i<=r; i++)
    res += s[i];
return res;

int solve(int l, int r)
return p[id][r]+q[id][l/k];

```

```

A. себе
#include <bits/stdc++.h>
#define pb push_back
#define all(x) (x).begin(), (x).end()
#define sz(x) (int)(x).size()
using namespace std;
typedef long long ll;
const int MAXN = (int)1e6 + 5;
const int MOD = (int)1e9 + 7;
int a[MAXN];
int d[MAXN];
int n;
int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cin >> n;
    for (int i = 1; i <= n; i++) { cin >> a[i]; }
    sort(a + 1, a + n + 1);
    if (a[n] <= 0)
        { cout << "1\n"; return 0; }
    int t = n;
    while (t > 2 && a[t-2] + a[t-1] > a[t] && a[t-1] > 0) { t--; }
    d[t] = (int)1e9;
    for (int i = 2; i <= n; i++)
        { d[i] = a[i] - a[i-1]; }
    vector <pair <int, int >> values;
    for (int i = 1; i <= n; i++)
        { values.pb({d[i], i}); }
    sort(all(values));
    set <int > S;
    for (int i = 1; i <= n; i++)
        { S.insert(i); }
    int ans = 0;
    int ptr = 0;
    for (int j = t; j <= n; j++)
        { while (ptr < sz(values) && values
          [ptr].first < a[j] { int idx = values
          [ptr].second;
          S.erase(idx);
          ptr++; }
        int i = *prev(S.lower_bound(j));
        ans += (j - i); }
    cout << ans << "\n";
    return 0; }

```

4-10-04

```

B.cpp
#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) 1e6 + 5;
// vector <vector<int> > where [MAXN];
vector <int> where [MAXN];
int a [MAXN];
int n;
int index - lim;
int fenw - max [MAXN];
int pref - max (int p)
{ int res = 0;
  for (; p > 0; p = p & - p) {
    res = max (res, fenw - max [p]);
  }
  return res;
}
void update - max (int p, int x) {
  for (; p <= n; p += p & - p) {
    fenw - max [p] = max (fenw - max [p], x);
  }
}
void add (int x) {
  int prv = 0;
  for (int pos : where [x]) {
    update - max (prv + 1, pos);
    prv = pos;
  }
  index - lim = min (index - lim, prv);
}
int main () {
  ios :: sync - with - stdio (0);
  cin . tie (0);
  cin >> n;
  for (int i = 1; i <= n; i++) {
    cin >> a [i];
    where [a [i]].pb (i);
  }
  index - lim = n;
}

```

```

for (int x = 0; x <= n - 1; x++)
  { add (x);
  int i = 1, ans = 0;
  while (i <= n) { int j;
    if (i > index - lim) {
      j = n + 1;
    }
    else {
      j = pref - max (i);
    }
    ans++;
    if (j == i) { ans--;
      break;
    }
    i = j;
  }
  cout << ans << "\n" [x == n];
  // curr << (double) clock () / clock_s -
  PER - SEC << endl; return 0;
}

```



```

C-еседі
// #pragma GCC optimize ("-Ofast")
#include <bits/stdc++.h>
#define pb push_back
#define all(x) (x).begin(), (x).end()
#define sz(x) (int)(x).size()
using namespace std;
typedef long long ll;
const int MAXN = (int)1e5 + 5;
const int K = 300;
const int l = MAXN / K + 2;
int P[i][MAXN], Q[i][i];
// S[MAXN], T[MAXN];
// a[MAXN], b[i];
int p[MAXN];
int n, m, q;
void sqrtUpd(int p, ll, x) {
    S[p] += x;
    T[p / K] += x;
}
void sqrtUpd(int id, int l, int r, int x) {
    int cl = l / K, cr = r / K;
    if (cl == cr) { for (int i = l; i <= r; i++) {
        P[id][i] += x;
    }
    return;
}
for (int i = l, j = (cl + 1) * K; i < j; i++) {
    P[id][i] += x;
}

```

```

// sqrtGet(int l, int r) {
    int cl = l / K, cr = r / K;
    // ret = 0;
    if (cl == cr) { for (int i = l; i <= r; i++) {
        ret += S[i];
    }
    return ret;
}
for (int i = 1; j = (cl + 1) * K; j < i; i++) {
    ret += S[i];
}
for (int i = cl * K; i <= r; i++) {
    ret += S[i];
}
return set;
}
int sqrtGet(int id, int l, int r) {
    return sqrtGet(id, r) - (l ? sqrtGet(id, l - 1); 0);
}
void build() { for (int i = 0; i * K <= n; i++) {
    int l = i * K, r = min(n, (i + 1) * K);
    for (int j = l; j * K <= n; j++) {
        for (int i = l; i <= r; i++) {
            Q[i][j] = Q[i][j - 1];
        }
    }
}
void update(int l, int r, int x) { int cl = l / K, cr = r / K

```

4-10-04

```

if (cl == cr) {
    for (int i = l; i <= r; i++)
        { sqrtUpd(p[i], x); }
    return; }
for (int i = l, j = (cl + 1) * K; i < j; i++)
    { sqrtUpd(p[i], x); }
for (int i = cr * K; i <= r; i++)
    { sqrtUpd(p[i], x); }
// query (int l, int r) {
//     res = sqrtGet(l, r);
//     for (int i = 0; i < l; i++) {
//         sqrtUpd(p[i], x);
//     }
//     for (int i = cl + 1; i <= cr; i++)
//         { b[i] += x; }
//     for (int i = cr * K; i <= r; i++) {
//         sqrtUpd(p[i], x);
//     }
//     // query (int l, int r) { // res += b[i] * sqrtGet(i, l, r); }
//     return res; }
int l = t * K, r = min(n, (t + 1) * K);
for (int i = l; i <= r; i++)
    b[i] = 0;

```

```

else if (tp == 2) {
    int l, r;
    cin >> l >> r;
    l--;
    r--;
    cout << query(l, r) << '\n';
}

```

```

else { int a, b;
    cin >> a >> b;
    a--;
    b--;
    exchange(a, b);
    return 0;
}

```

```

a) N = int((Input))
Sum = 0
For i in range(1, N+1):
    Sum I = 0
    For i in range(1, i+1):
        if i % 2 = 0:
            Sum / i + 1
        if Sum i % 2 = 0:
            sum += 1
            sum += 1
    print /sum
    
```

```

#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)1e6 + 5;
const int MOD = (int)1e9 + 7;
int a[MAXN];
int a[MAXN];
int n;
int main() {
    ios::sync_with_stdio(0);
    cin >> t;
    cin >> n;
    for(int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    sort(a + 1, a + n + 1);
    if (a[n] <= 0) {
        cout << "1\n";
        return 0;
    }
    int t = n;
    
```

```

while (t > 2 && a[t + 2] + a[t - 1] > a[t] && a[t - 1] > 0) {
    t--;
}
a[t] = (int)2e9;
for(int i = 2; i <= n; i++) {
    a[i] = a[i] - a[i - 1];
}
    
```

Парақтың артқы жағын толтырмаңыз / Обратную сторону листа не заполнять

a-еседі.

```

#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char* argv[]) {
    int i, n, a, max1, max2, f=0;
    cin >> n;
    cin >> max1;
    for (i=1; i<n; i++) {
        cin >> a;
        if (f==0) {
            if (a >= max1) { max2 = max1;
                max1 = a; }
            else max2 = a;
                f = 1; }
            else {
                if (a >= max1) {
                    max2 = max1;
                    max1 = a;
                }
                else
                    if (a > max2) max2 = a;
            }
        }
        cout << max2 << " ";
    }
}

```

B-еседі

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    cout << a * b * c * d;
}

```

c-еседі

```

return res;
void push(int b) {
    if (b/f) {
        return;
    }
    int l = l * k, r = min(n, (l+1) * k); for
    (int i = l; i < r; i++) {
        sgrt Upd(p[i]);
        b[b] = 0;
    }
    void exchange(int a, int b) {
        int ca = a/k, cb = b/k;
        if (ca != cb) {
            push(ca);
            push(cb);
            if (p[a] < p[b]) {
                sgrt Upd(ca, p[a], p[b]-1, -1);
                sgrt Upd(cb, p[a], -1, 1);
            }
        }
    }
}

```

с-сөзін.

} else {

```

    swap(p[a], p[b]);
    swap(p[b], p[a]-1);
    swap(p[a], p[b]);

```

int main() {

ios; sync_with_stdio(0); cin.tie(0);

cin >> n >> q;

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

```

    p[i];
    p[i];

```

}

build();

for (int i = 1; i <= q; i++) { int t;

cin >> t;

if (t == 1) {

int l, r;

cin >> l >> r >> a, t;

r--;

update(l, r, x);

} else if (t == 2) {

int l, r;

cin >> l >> r, t;

r--;

cout << query(l, r) << "\n";

} else {

int a, b;

cin >> a >> b, a--;

b--;

exchange(a, b); }

return 0;

4-11-01

```

1) #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)1e6 + 5;
   const int MOD = (int)1e9 + 7;
   int a[MAXN];
   int d[MAXN];
   int n;

   int main() {
       ios::sync_with_stdio(0);
       cin.tie(0);
       cin >> n;
       for (int i = 1; i <= n; i++) {
           cin >> a[i];
       }
       sort(a + 1, a + n + 1);
       if (a[n] <= 0) {
           cout << "1/n";
           return 0;
       }
       int t = n;
       while (t > 2 && a[t - 2] + a[t - 1] > a[t] &&
              a[t - 1] > 0) {
           t--;
       }
       d[1] = (int)2e9;
       for (int i = 2; i <= n; i++) {
           d[i] = a[i] - a[i - 1];
       }
       vector<pair<int, int>> values;
       for (int i = 1; i <= n; i++)
           s.insert(i);
   }

```

4-11-01

```
    int ans = 0;
    int ptr = 0;
    for (int j = 1; j <= n; j++) {
        while (ptr < sz(values) && values[ptr].first < a[j]) {
            int idx = values[ptr].second;
            s.erase(idx);
            ptr++;
        }
        int i = *prev(s.lower_bound(j));
        ans += (j - i);
    }
    cout << ans << "\n";
    return 0;
}
```

```
B) #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)1e6 + 5;
// Vector < vector < int >> where (MAXN);
int index_lim;
int fenw_max[MAXN];
int pref_max(int p) {
    int res = 0;
    for (; p > 0; p -= p & -p) {
        res = max(res, fenw_max[p]);
    }
    return res;
}

void add(int x) {
    int prv = 0;
    for (int pos : where[x]) {
        update_max(prv + 1, pos);
        prv = pos;
    }
    index_lim = min(index_lim, prv);
}

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

```

for (int i=1; i<=n; i++) {
    cin >> a[i];
    where[a[i]].pb(i);
}
index_lim = n;
for (int x=0; x<=n-1; x++) {
    add(x);
    int i=1, ans=0;
    while (i<=n) {
        int j;
        if (i > index_lim) {
            j = n+1;
        }
        else {
            j = pref_max(i);
        }
        ans++;
        if (j == i) {
            ans--;
            break;
        }
        cout << ans << " " [x==n];
    }
    // cerr << ans << " " [x==n];
    // cerr << (double) clock() / CLOCKS_PER_SEC << endl;
    return 0;
}

```

```

c) // #pragma GCC optimize ("-Ofast")
#include <bits/stdc++.h>
#define pb push_back
#define all(x) (x).begin(), (x).end()
#define sz(x) (int)(x).size()
using namespace std;
typedef long long ll;
const int MAXN = (int)1e5 + 5;
const int K = 300;
const int L = MAXN / K + 2;
int P[L][MAXN], Q[L][L];
// S[MAXN], T[MAXN];

```

Парақтың артқы жағын толтырмаңыз / Обратную сторону листа не заполнять

4-11-01

// a[MAXN], b[1];

int p[MAXN];

int n, m, q;

```
void sqrtUpd (int p, int x) {
  S[p] += x;
  T[p/K] += x;
}
```

```
} void sqrtUpd (int id, int l, int r, int x) {
  int cl = l/K, cr = r/K;
  if (cl == cr) {
    for (int i = l; i <= r; i++) {
      P[id][i] += x;
    }
    return;
  }
```

```
  for (int i = l, j = (cl+1)*K; i < j; i++) {
    P[id][i] += x;
  }
```

```
  for (int i = cl+1; i < cr; i++) {
    Q[id][i] += x;
  }
```

```
  for (int i = cr*K; i <= r; i++) {
    P[id][i] += x;
  }
}
```

```
} "sqrtGet (int l, int r) {
  int cl = l/K, cr = r/K;
  // ret = 0;
  if (cl == cr) {
    for (int i = l; i <= r; i++) {
      ret += S[i];
    }
    return ret;
  }
```

```
  for (int i = l, j = (cl+1)*K; i < j; i++) {
    ret += S[i];
  }
```

```
  for (int i = cl+1; i < cr; i++) {
    ret += T[i];
  }
```

```
  for (int i = cr*K; i <= r; i++) {
    ret += S[i];
  }
}
```

```

} return ret;

```

```

} int sqrtGet(int id, int p) {
    return P[id][p] + Q[id][p/k];
}

```

```

} int sqrtGet(int l, int r) {
    return sqrtGet(id, r) - (l ? sqrtGet(id, l-1); 0);
}

```

```

} void build() {
    for (int i=0; i * k < n; i++) {
        int l = i * k, r = min(n, (i+1) * k);
        for (int j=l; j < r; j++) {
            int x = p[j];
            int id = x / k;
            ++P[id][x];
            ++Q[id][id+1];
        }
    }
}

```

```

} void update(int l, int r, int x) {
    int cl = l / k, cr = r / k;
    if (cl == cr) {
        for (int i=l; i < r; i++)
            sqrtUpd(p[i], x);
    }
    return;
}

```

```

} for (int i=l, j=(cl+1) * k; i < r; i++)
    sqrtUpd(p[i], x);
}

```

4-11-01

```

} void push (int t) {
  if (! b[t]) {
    return;
  }
}

```

```

} int l = t * K, r = min (n, (t+1) * K);
  for (int i = l; i < r; i++) {
    sqrtUpd (p[i], b[t]);
  }
}

```

```

} b[t] = 0;
}

```

```

} void exchange (int a, int b) {
  int ca = a / K, cb = b / K;
  if (ca != cb) {
    push (ca);
    push (cb);
    if (p[a] < p[b]) {
      sqrtUpd (ca, p[a], p[b]-1, -1);
      sqrtUpd (cb, p[a], p[b]-1, 1);
    }
  }
}

```

```

} swap (p[a], p[b]);
}

```

```

} int main () {
  ios::sync_with_stdio (0);
  cin.tie (0);
  cin >> n >> q;
  for (int i = 0; i < n; i++) {
    cin >> p[i];
    p[i]--;
  }
  build ();
  for (int i = 1; i <= q; i++) {
    int tp;
    cin >> tp;
    if (tp == 1) {
      else if (tp == 2) {
        int l, r;
        cin >> l >> r;
        l--;
        r--;
        cout << query (l, r) << '\n';
      }
    }
    else {
      int a, b;
      cin >> a >> b;
      a--;
      b--;
      exchange (a, b);
    }
  }
}

```

```

}
} return 0;
}

```

```

a) N = int
sum = 0
for i in range(1, N+1):
    sum += i
    for j in range(1, i):
        sum += j
    if sum % 2 == 0:
        sum += 1
    else:
        sum += 1
    
```

```

print(sum)

```

```

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int MAXN = 1000000;
int a[MAXN];
int n;
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++)
        cin >> a[i];
    
```

```

if (n > 1);
for (int i = 1; i <= n; i++)
    cin >> a[i];
}
sort(a+1, a+n+1);
if (a[n] <= 0) {
    cout << "0";
    return 0;
}
int r = n;
    
```

```

while (3 <= r && a[r-2] + a[r-1] > a[r])
    r--;
}
a[r] = (int)2e9;
for (int i = 2; i <= n; i++)
    a[i] = a[i] - a[i-1];
}
vector<pair<int, int>> values;
for (int i = 1; i <= n; i++)
    
```

```

A
#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)1e6 + 5
const int MOD = (int)1e9 + 7
int a[MAXN];
int a[MAKIV];
int n;
int main() {
    ios::sync_with_stdio(0);
    cin >> t;
    cin >> n;
    for(int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    sort(a+1, a+n+1);
    if(a[n] <= 0) {
        cout << "1\n";
        return 0;
    }
    int t = n;

```

```

while(t > 2 && a[t-2] + a[t-1] > a[t] && a[t-1] > 0) {
    t--;
}
d[t] = (int)2e9;
for(int i = 2; i <= n; i++) {
    d[i] = a[i] - a[i-1];
}
vector<pair<int, int>> values;
for(int i = 1; i <= n; i++) {
    values.pb({d[i], i});
}
sort(all(values));
set<int> s;
for(int i = 1; i <= n; i++) {
    s.insert(i);
}
ll ans = 0;
int per = 0;
for(int j = t; j <= n; j++) {
    while(per < sz(values) && values[per].first < a[j]) {
        int idx = values[per].second;
        s.erase(idx);
        per++;
    }
}

```

0.

```

const int MAXN = (int)1e5 + 5;
const int K = 300;
const int l = MAXN / K + 2;
int p[MAXN], q[l][l];
ll s[MAXN], t[MAXN];
ll a[MAXN], b[l];
int p[MAXN];
int n, m, q;
void sqrtUpd(int p, ll x) {
    s[p] += x;
    t[p/K] += x;
}
}
void sqrtUpd(int id, int l, int r, int x) {
    int cl = l/K, cr = r/K;
    if (cl == cr) {
        for (int i = l; i <= r; i++) {
            p[id][i] += x;
        }
    }
    return;
}
for (int i = 1; i <= l; i++) {
    for (int j = (cl+1)*K; j < i; j++) {
        p[id][j] += x;
    }
}
for (int i = 1; i <= l; i++) {
    for (int j = (cl+1); j <= cr; j++) {
        q[id][j] += x;
    }
}

```

```
}
```

```
}
```

```
// sqrt Getl (int l, int r) {
```

```
    int d = l / k, cr = r / k
```

```
    // ret = 0;
```

```
    if (d == cr) {
```

```
        for (int i = l; i <= r; i++) {
```

```
            ret += S[i];
```

```
        }
```

```
        return ret
```

```
    }
```

```
    for (int i = l, i = (l + 1) * k; i < j; i++) {
```

```
        ret += S[i];
```

```
    }
```

```
    for (int i = d + 1; i <= cr; i++) {
```

```
        ret += T[i];
```

```
    }
```

```
    for (int i = cr * k; i <= r; i++) {
```

```
        ret += S[i]
```

```
    }
```

```
    return ret;
```

```
}
```