

```
using namespace std;
typedef long long ll;
const int MAXN = (int)1e6 + 5;
const int MOD = (int)1e9 + 7;
```

```
ans += (j-1);
cout << ans << '\n';
return 0;
```

```
B. using namespace std;
typedef long long ll;
const int MAXN = (int)1e6 + 5;
// vector <vector<int>>
```

```
int a[MAXN];
int d[MAXN];
int n;
int main ()
```

```
{ ios::sync_with_stdio()
```

```
cin >> n;
```

```
int main ()
```

```
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] && 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++)
```

```
{ values.pb({d[i], i});
```

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

```
set<int> s;
```

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

```
{ s.insert(i);
```

```
int ans = 0;
```

```
int ptr = 0;
```

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

```
{ while (ptr < s.size() && values[ptr].first < a[i])
```

```
{ int idx = values[ptr].second;
```

```
{ erase(idx);
```

```
ptr++;
```

```
int j = ptr < s.size() ? s.lower_bound(i) : 1;
```

Парақтың артқы жағын толтырмаңыз / Обратную сторону листа не заполнять

B. Using namespace & col

$i \in (j) \Rightarrow i$

```

type def long long ll;
const int MAXN = (int) 1e6 + 5;
// vector <vector<int> where (MAXM;
vector<int> where [MAXN]
int a [MAXN];
int n;
int index_lim;
int temp_max [MAXN]
int pref_max (int p)
{ int res = 0;
  for (i, p; i <= p; i++)
  { res = max(res, temp_max [i]);
  }
  return res;
}
void update_max (int p, int x)
{ for (i, p; i <= n; i++)
  { temp_max [i] = max(temp_max [i], x);
  }
}
void add (int x)
{ int p = 0;
  for (int pos: where [x])
  { update_max (p + 1; pos);
    p = pos;
  }
  index_lim = min (index_lim, p);
}
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; x++)
  { add (x);
    int i = 1, ans = 0;
    while (i <= n)
    { int j = i;
      while (j <= n)
      { int i;
        if (ans >= i)
      }
    }
  }
}

```

```

ans = -1;
break;
i = j;
cout << ans << "\n" << "[x = n]";
cerr << (double) clock () / CLOCKS_PER_SEC
<< endl; return;
}

```