

## Baekjoon / 수 고르기 (2230번)

### 문제

$N$ 개의 정수로 이루어진 수열  $A[1], A[2], \dots, A[N]$ 이 있다. 이 수열에서 두 수를 골랐을 때(같은 수일 수도 있다), 그 차이가  $M$  이상이면서 제일 작은 경우를 구하는 프로그램을 작성하시오. 예를 들어 수열이  $\{1, 2, 3, 4, 5\}$ 라고 하자. 만약  $M = 3$ 일 경우, 1 4, 1 5, 2 5를 골랐을 때 그 차이가  $M$  이상이 된다. 이 중에서 차이가 가장 작은 경우는 1 4나 2 5를 골랐을 때의 3이 된다.

### 입력

첫째 줄에 두 정수  $N, M$ 이 주어진다. 다음  $N$ 개의 줄에는 차례로  $A[1], A[2], \dots, A[N]$ 이 주어진다.

### 출력

$M$  이상이면서 가장 작은 차이를 출력한다. 항상 차이가  $M$  이상인 두 수를 고를 수 있다.

### 제한

- $1 \leq N \leq 100,000$
- $0 \leq M \leq 2,000,000,000$
- $0 \leq |A[i]| \leq 1,000,000,000$

### 예제 입력 1

```
3 3
1
5
3
```

### 예제 출력 1

```
4
```

### 문제 접근 방법

1. 수열  $A$ 를 오름차순으로 정렬합니다.
2. 투 포인터 (two pointer) 알고리즘을 사용합니다.
3. 두 포인터  $start, end$ 를 초기화하고 다음을 반복합니다.
  - $A[end] - A[start] < M$ 이면  $end$ 를 증가시킵니다.
  - 그렇지 않으면, 현재 차이를 정답 후보로 저장하고  $start$ 를 증가시킵니다.
4. 모든 경우를 검사하여  $M$  이상인 최소 차이를 출력합니다.

## 코드

```
// 2230 / 수 고르기 / baekjoon
// 64ms, 40m

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int N;
    long long M;
    cin >> N >> M;

    vector<long long> A(N);
    for (int i = 0; i < N; ++i) cin >> A[i];

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

    int start = 0, end = 0;
    long long answer = 2e9 + 1;

    while (start < N && end < N) {
        long long diff = A[end] - A[start];

        if (diff < M) {
            end++;
        } else {
            answer = min(answer, diff);
            start++;
            if (start > end) end = start;
        }
    }

    cout << answer << '\n';
    return 0;
}
```