//algoleague

Soulmate

⌂ • Contest List • Algorithm Competition Summer Camp 2023 Foundation Upsolving Contest • Problem List • Soulmate • Problem

Problem

Submissions

Discussion Coming Soon

Two different elements **(not necessarily different integers)** in an array are considered *soulmates* if their sum is divisible by P.

Given an array A consisting of n integers, can you determine whether there exist any **soulmates**?

P.S. You may try to make your $\mathcal{O}(n^2)$ solution pass with some small tweaks but we advise you not to do so (even if the test cases are hackable somehow). Have you ever heard about any powerful data structures?

Input Format

The first line contains two integers n and P.

The following line contains n space-separated integers, the elements in A:

$A_1\ A_2\ A_3 \cdots A_n$

Output Format

Print `Yes` if there exist soulmates, else, print `No`.

Be aware that your answer will be checked case-sensitively, i.e. do not print `yes`, `YES` etc.

Constraints

- $2 \leq n \leq 5 \cdot 10^5$
- $|A_i| \leq 10^9$
- $1 \leq P \leq 10^9$

Sample Input 1

```
3 999
0 1 -2
```

Sample Output 1

```
No
```

Sample Input 2

```
2 2
2 3
```

Sample Output 2

```
No
```

Explanation 2

Sum of the only pair is 5, which is not divisible by 2.

Sample Input 3

```
5 9
-1 -2 -3 -4 -5
```

Sample Output 3

```
Yes
```

Explanation 3

Only -4 and -5 are soulmates, their sum -9 is divisible by 9.

```cpp
#include<bits/stdc++.h>
using namespace std;

int main() {

    int n, P;
    cin >> n >> P;

    unordered_map<int, int> modCount;

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

        int mod = ((a % P) + P) % P;

        if (modCount.count((P - mod) % P)) {
            cout << "Yes\n";
            return 0;
        }

        modCount[mod]++;
    }

    cout << "No\n";
    return 0;
}
```

C++ (GCC 9.2.0)   Bright

Memory Limit (kB) : 256000   Time Limit (s) : 1

Upload File   Test against custom test case   Run Code   Submit

✓ Sample Test Case 0
✓ Sample Test Case 1
✓ Sample Test Case 2

Accepted

Input(stdin)
```
2 2
2 3
```

Output(stdin)
```
No
```

Expected Output
```
No
```