Non-Divisible Subset

Given a set of distinct integers, print the size of a maximal subset of S where the sum of any 2 numbers in S' is *not* evenly divisible by k.

Input Format

The first line contains 2 space-separated integers, n and k, the number of values in S and the *non* factor.
The second line contains n space-separated integers describing $S[i]$, the unique values of the set.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq k \leq 100$
- $1 \leq S[i] \leq 10^9$
- All of the given numbers are distinct.

Output Format

Print the size of the largest possible subset (S').

Sample Input

```
4 3
1 7 2 4
```

Sample Output

```
3
```

Explanation

The sums of all permutations of two elements from $S = \{1, 7, 2, 4\}$ are:

```
1 + 7 = 8
1 + 2 = 3
1 + 4 = 5
7 + 2 = 9
7 + 4 = 11
2 + 4 = 6
```

We see that only $S' = \{1, 7, 4\}$ will not ever sum to a multiple of $k = 3$.

Author zxqfd555
Difficulty Medium
Max Score 20
Submitted By 36867

NEED HELP?
View discussions
View editorial
View top submissions

RATE THIS CHALLENGE

MORE DETAILS
Download problem statement
Download sample test cases
Suggest Edits

Current Buffer (saved locally, editable) Java 7

```
10 ▼ import java.io.*;
11   import java.math.*;
```

```java
import java.text.*;
import java.util.*;
import java.util.regex.*;

public class Solution {

    /*
     * Complete the nonDivisibleSubset function below.
     */
    static int nonDivisibleSubset(int k, int[] S) {
        /*
         * Write your code here.
         */

    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        String[] nk = scanner.nextLine().split(" ");

        int n = Integer.parseInt(nk[0].trim());

        int k = Integer.parseInt(nk[1].trim());

        int[] S = new int[n];

        String[] SItems = scanner.nextLine().split(" ");

        for (int SItr = 0; SItr < n; SItr++) {
            int SItem = Integer.parseInt(SItems[SItr].trim());
            S[SItr] = SItem;
        }

        int result = nonDivisibleSubset(k, S);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedWriter.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code