Let's call product(x) the product of x's digits. Given an array of integers a, calculate product(x) for each x in a, and return the number of distinct results you get.

Example

For a = [2, 8, 121, 42, 222, 23], the output should be  
uniqueDigitProducts(a) = 3.

Here are the products of the array's elements:

* 2: product(2) = 2;
* 8: product(8) = 8;
* 121: product(121) = 1 \* 2 \* 1 = 2;
* 42: product(42) = 4 \* 2 = 8;
* 222: product(222) = 2 \* 2 \* 2 = 8;
* 23: product(23) = 2 \* 3 = 6.

As you can see, there are only 3different products: 2, 6 and 8.

Input/Output

* **[execution time limit] 3 seconds (cs)**
* **[input] array.integer a**

*Guaranteed constraints:*  
1 ≤ a.length ≤ 105,  
1 ≤ a[i] ≤ 109.

* **[output] integer**
  + The number of different digit products in a.

<https://app.codesignal.com/arcade/code-arcade/sorting-outpost/oY6FASrCMEqkxwcAC/description>

static int Product(int n)

{

int prod = 1;

while (n > 0)

{

prod \*= (n % 10);

n /= 10;

}

return prod;

}

static int uniqueDigitProducts(int[] a)

{

HashSet<int> hash = new HashSet<int>();

for (int i = 0; i < a.Length; i++)

{

hash.Add(Product(a[i]));

}

return hash.Count;

}