```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:45 CST 2018*/

/*Given a non-negative integer represented as a non-empty array of digits, plus one to the integer.

        You may assume the integer do not contain any leading zero, except the number 0 itself.

        The digits are stored such that the most significant digit is at the head of the list.*/

import java.util.Arrays;
import java.lang.Math;
import java.lang.System;
import java.util.HashMap;
import java.lang.Integer;
import java.util.Iterator;
import java.util.Map;


public class Main {

    public static void main(String[] args) {
        int[] nums = {2, 2};

        Solution solution = new Solution();
        int receive = solution.findShortestSubArray(nums);


        System.out.println("haha");
    }

}


class Solution {
    public int[] plusOne(int[] digits) {

        int digitsLength = digits.length;
        if (digits == null || digitsLength < 1) {
            return digits;
        }

        int additionBit = 0;
        int currentSum = digits[digitsLength - 1] + 1;
        if (currentSum == 10) {
            additionBit = 1;
            digits[digitsLength - 1] = 0;
        } else {
            digits[digitsLength - 1] = currentSum;
        }

        for (int i = digitsLength - 2; i >= 0; i--) {
            currentSum = digits[i]+additionBit;
            if (currentSum==10){
                additionBit = 1;
                digits[i] = 0;
            } else{
                additionBit = 0;
                digits[i] = currentSum;
            }
        }

        if (additionBit==0){
            return digits;
        } else {
            return plusOneHelper(digits);
        }
    }

    private int[] plusOneHelper(int[] digits){

```

```
        int digitsLength = digits.length;

        int[] returnValue = new int[digitsLength+1];

        returnValue[0] = 1;
        for (int i = 1; i <= digitsLength;i++){
            returnValue[i] = digits[i-1];
        }

        return returnValue;
    }
}
```