

Program for Prime Subtraction Operation

```
public class PrimeSubtraction {
    public static boolean primeSubOperation(int[] nums) {
        // Create an array to store prime numbers up to 1000
        int[] prime = new int[1001];

        // Sieve of Eratosthenes to find prime numbers up to 1000
        for (int i = 2; i <= Math.sqrt(1000); i++) {
            if (prime[i] == 0) {
                for (int j = i * i; j <= 1000; j += i) {
                    prime[j] = 1;
                }
            }
        }

        // Iterate through the 'nums' array from right to left
        for (int i = nums.length - 2; i >= 0; i--) {
            if (nums[i] < nums[i + 1]) {
                continue; // If current element is smaller than next, skip
            } else {
                int num = nums[i];
                int temp = 2;

                while (num >= nums[i + 1]) {
                    if (temp == 1001) {
                        return false; // If we exceed 1000, return false
                    }
                    if (num < 1) {
                        return false; // If num becomes negative, return false
                    }

                    num = nums[i];
                    if (prime[temp] == 0) {
                        num = num - temp; // Subtract prime number
                    }

                    if (num < 1) {
                        return false; // If num becomes negative, return false
                    }

                    temp++;
                }

                nums[i] = num; // Update the value in the array
            }
        }
        return true;
    }

    public static void main(String[] args) {
```

```
int[] nums1 = {4,9,6,10};

boolean result1 = primeSubOperation(nums1);

System.out.println("Result : " + result1);
    }
}
```

OUTPUT

Result : true

Program for Count the Digits That Divide a Number

```
public class CountDivisibleDigits {  
  
    public static int countDigits(int num) {  
        int count = 0;  
        int digit = 0;  
        int temp = num;  
  
        // Iterate through each digit of the number  
        while (num != 0) {  
            digit = num % 10; // Get the rightmost digit of the number  
  
            // Check if the rightmost digit is not zero and divides the original  
            number evenly  
            if (digit != 0 && temp % digit == 0) {  
                count++; // Increment the count if the digit is divisible  
            }  
            num = num / 10; // Move to the next digit (remove the rightmost  
            digit)  
        }  
  
        return count;  
    }  
  
    public static void main(String[] args) {  
        int number = 1248;  
        int count = countDigits(number);  
        System.out.println("Result : " + count);  
    }  
}
```

OUTPUT

Result : 4