

Buscando primos

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

Debes escribir una función para buscar números primos en una cadena de N dígitos decimales (desde 0 hasta 9).

Un número entero positivo se dice primo, si tiene exactamente dos divisores (que serán necesariamente 1 y el propio número). Notar que 1 tiene un solo divisor y por lo tanto no es primo. Los primeros números primos son $2, 3, 5, 7, 11, 13, \dots$.

Concretamente, para cada entero k entre 1 y 4 inclusive, deberás contar la cantidad de subcadenas de k dígitos consecutivos que existen en la cadena principal de N dígitos, tales que los k dígitos formen un número primo. Notar que una subcadena con **ceros a la izquierda**, como por ejemplo 007 , **no forma** un número primo.

Descripción de la función

Debes implementar la función `buscaprimos(s)`, que recibe una cadena de texto s de longitud N , formada exclusivamente por dígitos decimales.

La función debe retornar un arreglo de exactamente 4 enteros. El k -ésimo de ellos (contando desde 1) deberá indicar cuántas subcadenas de s k dígitos consecutivos de s existen, que forman un número primo.

Evaluador

El evaluador local lee de la entrada estándar una única cadena de texto s .

Escribe en la salida estándar los números del arreglo retornado por la función.

Restricciones

- $1 \leq N \leq 1.000$
- s solo contiene dígitos, de 0 a 9 inclusive

Ejemplos

Si la entrada fuera:

```
345892304534543
```

La salida correcta será:

```
8 4 0 1
```

Puntuación

Se recibe el 25 % del puntaje por tener el primer número correcto.

Se recibe el 50 % del puntaje por tener los primeros dos números correctos.

Se recibe el 75 % del puntaje por tener los primeros tres números correctos.

Subtareas

1. $N = 1$ (4 puntos)
2. $N = 2$ (4 puntos)
3. $N = 3$ (8 puntos)
4. $N = 4$ (12 puntos)
5. $N = 5$ (20 puntos)
6. Sin más restricción (52 puntos)