



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

### Diseño y Análisis de Algoritmos - IIC2283

#### Tarea 3

Entrega: lunes 21 de noviembre

En esta tarea usted deberá implementar en Java o Python la función **TestPrimalidad**( $n, k$ ) vista en clases. En su implementación deberá tomar en cuenta lo siguiente:

- El primer argumento de **TestPrimalidad** es un número natural  $n \geq 2$  con un número arbitrario de dígitos. Para representar este tipo de números debe crear una clase **Entero** que utilice un arreglo para almacenar los dígitos de un número.
- Debe implementar todas las funciones utilizadas por **TestPrimalidad** a partir de las funciones básicas del lenguaje de programación utilizado (no utilice librerías específicas para manejar números enteros grandes). Por ejemplo, debe implementar funciones que permitan calcular  $a \cdot b$  y  $a \bmod b$ , donde  $a$  y  $b$  son dos objetos de tipo **Entero**. Nótese que sus implementaciones deben utilizar la representación de los enteros grandes, vale decir, deben operar sobre arreglos de dígitos.
- El segundo argumento de **TestPrimalidad** es un número natural  $k$  de unos pocos dígitos (vamos a considerar  $k \leq 200$ ), por lo que no necesita utilizar la clase **Entero** para representar este número.

Para mostrar que su programa funciona correctamente se deberá proveer de una interfaz que de tres opciones al usuario:

1. Leer  $n$  y  $k$  desde pantalla, y mostrar el resultado de la llamada **TestPrimalidad**( $n, k$ ).
2. Leer desde un archivo una lista  $n_1, k_1, n_2, k_2, \dots, n_\ell, k_\ell$  y escribir en un archivo los resultados de las llamadas **TestPrimalidad**( $n_1, k_1$ ), **TestPrimalidad**( $n_2, k_2$ ),  $\dots$ , **TestPrimalidad**( $n_\ell, k_\ell$ ). En este caso se debe preguntar al usuario los nombres de los archivos de entrada y salida, y se puede suponer que los números en el archivo de entrada vienen separados por una coma y un espacio. Por ejemplo, el siguiente es un posible archivo de entrada:

```
13, 100, 641, 100, 809734092839840293480393450509333429838475869484985096049504,  
20, 2431, 100
```

En el archivo de salida en este caso debe aparecer lo siguiente:

```
PRIMO, PRIMO, COMPUESTO, COMPUESTO
```

3. Salir de la ejecución del programa.

En la implementación de la interfaz puede suponer que las entradas vienen en el formato correcto (por ejemplo, no se dará como entrada un string cuando se espera que la entrada sea un número).