

## Control 11

### Contar “sumdivisibles”

El objetivo de este control es familiarizarse con el diseño de algoritmos recursivos basados en el esquema de “vuelta atrás”.

#### **El problema**

Se dice que un entero positivo es “sumdivisible” si la suma de sus dígitos es divisible por el número de dígitos y al quitar el último, el resultado es también “sumdivisible”. Implementa un algoritmo de “vuelta atrás” que tomando como entrada (i) un dígito positivo **D**; (ii) un entero positivo **K**, devuelva la cantidad total de números “sumdivisibles” de **K** dígitos que comienzan por **D**.

#### **Trabajo a realizar**

Para realizar el control se proporciona un archivo `control11.cpp` que contiene un programa que lee por la entrada estándar líneas que contienen, por este orden: (i) el dígito **D** por el que deben comenzar los números, y (ii) el número de dígitos **K** que dichos números deben tener. Después de leer cada línea, invoca al subprograma `num_sumdivisibles`, y escribe por la salida la cantidad de números sumdivisibles de **K** dígitos cuyo primer dígito es **D**.

El final de la entrada se indica mediante una línea con **0** como único valor.

Ejemplos de entrada / salida:

Entrada	Salida
1 2	5
2 3	16
5 8	112
9 20	150
0	

El trabajo a realizar consiste en implementar el algoritmo (como cuerpo de la función `num_sumdivisibles`), así como en justificar su diseño, contestando a cada uno de los apartados que aparecen entre comentarios en la plantilla entregada.

#### **Importante:**

- Únicamente se evaluarán aquellas entregas que superen satisfactoriamente los casos de prueba del juez.
- No modificar el código proporcionado. Únicamente debe implementarse el subprograma `num_sumdivisibles`, **junto con los subprogramas auxiliares que se consideren oportunos.**
- **No se corregirá ninguna entrega en la que no se hayan incluido los nombres de los miembros del grupo que han realizado el trabajo en el comentario que se incluye al comienzo del archivo.**