

# Ejercicio 1

Dado el archivo ejercicio1.h realiza los siguientes ejercicios.

- 1) Programa en el ejercicio1.c las funciones ahí declaradas:
  - a) `showArray`: Recorre un array e imprímelo por pantalla.
  - b) `setArray`: Setea un array mediante un bucle de 0 a MAX, siendo MAX un valor que se le pasa por parámetro. El valor que se le da al array es el resultado fórmula  $i*n$ , siendo  $n$  cada iteración del bucle.
  - c) `vectorDotVectorAndMap`: Multiplica cada posición de dos vectores y lo mapea y se almacena en out el resultado. Los dos vectores tienen que tener el mismo tamaño. Si no se almacena en outErr un vector, de tamaño uno, con el valor 0.
  - d) `numDotVectorAndMap`: Multiplicación escalar. Multiplica un vector por un número y lo mapeas en un nuevo vector.
- 2) Implementar el main.
  - a) Se setean dos vectores `res1`, `resErr`, `in1`, `in2` e `in3` usando la función `setArray`. El tamaño del vector `out`, `in1` e `in2` va a ser el marcado por `Size`, 100. `in3` va a tener tamaño 200. `out_size_1` un vector de tamaño 1.
  - b) Se realiza `numDotVectorAndMap` de `in1` y 50, y se almacena en la variable llamada `res1`.
  - c) Imprime `out` con `showArray`.
  - d) Programa `vectorDotVectorAndMap` de `in1` y `in2`, y se almacena en una variable llamada `res1`.
  - e) Imprime `res1` con `showArray`.
  - f) Programa `vectorDotVectorAndMap` de `in2` e `in3`, y se almacena en una variable llamada `resErr`.
  - g) Imprime `resErr` con `showArray`.
- 3) Mide los tiempos de ejecución de cada variable y apúntalo en una tabla. Prueba los diferentes niveles de optimización.
- 4) Utilizar la keyword `restrict` y comparar los ciclos con y sin ella, con los diferentes niveles de optimización.