

1. Escribe un programa en ensamblador ARM que dado un vector de números enteros (con número de elementos **numel**), escriba en **resultado** la diferencia entre el valor máximo y mínimo del vector. En el ejemplo, resultado = máximo(vector)-mínimo(vector) = 9 - (-7) = 16.

El fichero con el programa **debe llamarse ej1.s**

```
.data
numel:      .word 8
vector:     .word 8,-3,4,-7,9,-7,6,-1
resultado:  .word 2

.text
.global main
main:
    ...
```

El código debe funcionar para cualquier instancia de datos. Por ejemplo:

```
numel:      .word 2
vector:     .word 3, 3
```

En este caso, resultado = máximo(vector)-mínimo(vector) = 3 - (3) = 0.

```
numel:      .word 0
vector:     .word 0
```

En este caso, resultado = 0

```
numel:      .word 5
vector:     .word -1,-2,-3,-4,-5
```

En este caso, resultado = máximo(vector)-mínimo(vector) = (-1) - (-5) = 4.

2. Escribe una **función contperfect** en ensamblador ARM. Dado un vector con números enteros no-negativos que termina en un número negativo. La función debe determinar cuántos (contar) de los números son cuadrados perfectos. La dirección del vector se pasa por el registro r0 y el resultado también se pasa por el registro r0.

Para probar tu función puedes usar el código que aparece a continuación y usar tu función URperfect. No olvides que se trata de un ejemplo, y el programa main podría ser cualquier otra con su propio (cualquier) contenido de registros y memoria (seguir el convenio).

El fichero con el programa **debe llamarse ej2.s**

```
.data
vector:      .word 9,5,4,10,7, -1
res:         .word 9

.text
.global main
main:        ldr r0, =vector
             push {lr}
             bl contperfect
             pop {lr}
             ldr r4,=res
             str r0,[r4]
             bx lr

contperfect: @codigo a desarrollar en este ejercicio

             bl URperfect
```