**Task 1: Multiplication and sum of array elements**

Write an ARMv8 assembly program that multiplies each element of an integer array by a given factor, then calculates the sum of the elements of the resulting array. The program should use functions to achieve these tasks.

**Functions:**

1. **MultiplyArrayElements:** This function should take three arguments: the base address of the array, the total number of elements in the array, and the multiplication factor. It multiplies each element of the array by the factor.
2. **SumArrayElements:** This function calculates the sum of all elements in the array after they have been multiplied. It takes two arguments: the base address of the modified array and the total number of elements.

**Example:**

Given the array [1, 2, 3, 4, 5] and a multiplication factor of 2, the program should output the sum of the array [2, 4, 6, 8, 10], which is **30.**

.global \_start

.section .data

array: .quad 1, 2, 3, 4, 5

.section .text

\_start:

// Your code

end\_prog:

mov x8, #93 // Exit syscall number

mov x0, #0 // Exit status

SVC #0

**Task 2: Reverse array and compare of array elements**

**Functions:**

1. **ReverseArray:** This function should take two arguments: the base address of the array and the total number of elements in the array. It reverses the order of the array elements in place (meaning that after the function completes, the original array stores its elements in reverse order).
2. **CompareArrays**: This function compares the original array with its reversed version to check if the array is symmetrical. It takes three arguments: the base address of the original array, the base address of the reversed array (which can be the same as the original array if reversed in place), and the total number of elements. The function returns true (or a non-zero value) if the array is symmetrical, and false (or zero) otherwise.

**Example:**

Given the array [1, 2, 3, 2, 1], the program should indicate that the array is symmetrical.

For an array [1, 2, 3, 4, 5], the program should indicate that the array is not symmetrical.

# steps of compilation in assembly

# step-1: compile

**as -o filename.o filename.s**

# step-2: Link

**ld -o programname filename.o**

# step-3: run the compiled program

**./programname**

# Compile the assembly code using C compiler

**gcc -o program program.s**