ARM Assembly Language programs - Set2

1. Add a series of 16-bit numbers stored in sequential location in memory (called Table) and store the result in memory.
   Skeleton:

```
*Add a series of 16-bit numbers
        AREA Program, CODE, READONLY
        ENTRY
;Write the logic here

Table DCD &2040
        DCD &1C22
        DCD &0242
        DCD &0001
TableEnd DCD &0
Length DCW (TableEnd - Table) / 4
        ALIGN

        AREA Data, Data, READWRITE
Result DCD 0
```

2. Examine a list of items stored in a memory location for a match with a new item. If the new item is there in the list, end the code. If the new item is not in the list store the new item at the end of the list. (Note : Keep the list items in ROM first. Copy from ROM to RAM and then work on that)

```
        AREA Program, CODE, READONLY
        ENTRY
 ;Your logic
NewItem DCD &16FA

Start DCD &4       ;length of list
      DCD &5376 ;items
       DCD &7615
       DCD &138A
       DCD &21DC

        AREA Data1, DATA, READWRITE
List  DCD  0
```

3. Find the GCD of two numbers (Euclid algorithm) with and without using Conditional Execution. Example for conditional execution instructions that can be used are SUBLT, SUBHI . Algorithm to find the GCD is

```
function gcd(a, b)
        if a = 0
                return b
        while b ≠ 0
                if a > b
                        a := a − b
                else
                        b := b − a
        return a
```