Assignment 1

Michelle Weber, Sanja Janevska

12.09.2025

Task 1
Write the following code in the simulator and run it:

```
MOVZ        X0, #5
MOVZ        X1, #10
ADDI        X1, x1, #2
ADD         X2, X0, X1
```

After the program was run, the register X2 contains the DEC value 17.

Task 2
Translate the following machine code instructions to LEGv8 Assembly code:

11010010100000000000100000000010
Translates to: MOVZ X2, #128

11010010100000000001110011100100
Translates to: MOVZ X4, #231

11001011000000100000000010000101
Translates to: SUB X5, X4, X2

D360 0CA5
Translates to: 11010011011000001100
Translates to: LSL X5, X5, X0, #3

Task 3
Create a LEGv8 Assembly program to calculate the value of the following expression:

$$4 * 5 + 16 * 11 + 25$$

When finished, the result shall be stored in register x0.

```
MOVZ x9, #25
MOVZ x10, #11
LSL x10, x10, #4
ADD x9, x9, x10
MOVZ x10, #5
LSL x10, x10, #2
ADD x0, x9, x10
```

Task 4
Write a LEGv8 Assembly program to calculate the sum 1 893 423 + 443 924. The numbers are decimal integers.

```
MOVZ          x0, #59169
LSL           x0, x0, #5
MOVZ          x9, #15
ADD           x0, x0, x9
MOVZ          x9, #27745
LSL           x9, x9, #4
MOVZ          x10, #4
ADD           x9, x9, x10
ADD           x0, x0, x9
```

Task 5
Write a LEGv8 Assembly program to calculate the sum

$$1 + 3 + 5 + ... + 99.$$

When finished, the sum shall be stored in register x1.

```
MOVZ          x1, #1
MOVZ          x9, #3
MOVK          x10, #100
MOVZ          x11, #2
loop:    CMP    x9, x10
         B.GE end
         ADD    x1, x1, x9
         ADD    x9, x9, x11
         B loop
end:
```

Task 6

```
//Set up base memory address
MOVZ          x7, #0x1000, LSL #16
//Store the numbers 1, 4, 1, 5, 9, 2 in dynamic memory
MOVZ          x1, #1
STUR          x1, [x7, #0]
MOVZ          x1, #4
STUR          x1, [x7, #8]
MOVZ          x1, #1
STUR          x1, [x7, #16]
MOVZ          x1, #5
STUR          x1, [x7, #24]
MOVZ          x1, #9
STUR          x1, [x7, #32]
MOVZ          x1, #2
STUR          x1, [x7, #40]
```

Write a loop to add all the numbers stored in memory. When finished, the result shall be stored in register x0.

```
MOVZ          x8, #6
MOVZ          x9, #8
MOVZ          x10, #1
loop:
   LDUR         x1, [x7, #0]
   ADD          x0, x0, x1
   ADD          x7, x7, x9
   SUBS         x8, x8, x10
   B.NE         loop
```