VNR VJIET

Name of the Laboratory: ESD Laboratory.

Name of the Experiment:

Experiment No: 02 **Date:** 24/2/23

① Write a program to add first 10 numbers.

Program:

```
        AREA  SUMNUM, CODE, READONLY
        ENTRY
        EXPORT __main

__main
        MOV   R0, #10
        MOV   R1, #00
 LOOP
          ADD   R1, R0.
          SUBS  R0, #1
          BNE   LOOP
          LDR   R2, = sum.
          STR   R1, [R2]
STOP B  STOP.
          AREA  SUMNUM, DATA, READWRITE
 SUM  DCD  0X0
          END.
```

o/p:

```
R0  :  0x0.
R1  :  0x00000037.
R2  :  0x00000118.
```

(2) Write a program to find the sum of first n natural numbers without loop.

Program :

```
    AREA  NSUM, CODE, READONLY
    EXPORT __main

__main

          MOV    R0, #10
          MOV    R5, #2
          ADD    R1,R0, #1
          MUL    R3, R1, R0
          UDIV   R7, R3, R5
          LDR    R6, =SUMM
          STR    R7, [R6]
          AREA   DATA2, DATA, READWRITE
SUMM  DCD   0x0
          END.
```

Output :

```
R0    0x0000000A
R1    0x0000000B
R3    0x0000006C
R5    0x00000002
R6    0x10000000
R7    0x00000037.
```

Memory

```
0x10000000 : 00000037
```

(8) Write an assembly program to multiply to 32-bit integers.

PROGRAM :

```
        area multiplication, Code, ReadOnly
        entry
        EXPORT __main
NUM1    DCD   0x4587654
NUM2    DCD   0x3456985
result1 DCD   0x0
result2 DCD   0x0

__main
        LDR   R1, NUM1
        LDR   R2, NUM2
        UMULL R4, R3, R2, R1
        LDR   R5, =result1
        LDR   R6, =result2
        str   R3, [R5]
        str   R4, [R6]
        AREA  DATA2, DATA, READWRITE
        END
```

O/p :

```
                              → Decimal
R1 → 0x04587654   (72906324)
R2 → 0x03456985   (54880645)
R3 → 0x000E3705   (Lower)
R4 → 06B21EDA4    (Higher)
R5 → 0x108
R6 → 0x10C
```

Result :

E3705 6B21 EDA4

(4) Write a program to find factors of a number.

PROGRAM :

```
        area fact, Code, ReadOnly.
        entry
        EXPORT __main
fact DCD  0X0
__main
        mov  R1, #1
        mov  R2, #5
loop
        MUL  R1,R2,R1
        SUBS  R2, #1
        bne loop
        LDR  R4, = fact
        STR  R3, [R4]
        AREA  DATA 2, DATA, ReadWrite
        END.
```

o/p :

Final iteration values.

```
R1  :  0x00000078.  (120) -> decimal value.
R4  :  0X 00000100.
```

Address: R4

```
0X00000100  :  00000000 0101F04F 0205F04F F101FB02 D1FB3A01
               60234C00  00000100.
```

VNR VJIET

Name of the Laboratory: ESD laboratory

Name of the Experiment:

Experiment No: 03 **Date:** 3/3/23

① Write an Assembly language program to add an array of 16 bit numbers and store the result in 32-bit register.

Program :

```
        AREA  ADD16, CODE, READONLY
        EXPORT  __main
__main
        LDR   R0, = count
        LDR   R1, = ARRAY16
        MOV   R2, 0X0
        LDR   R3, [R0]
LOOP
        LDRH  R4,[R1], #4
        ADD   R2,R2, R4
        SUBS  R3,R3, #1
        BNE   LOOP
        STR   R2, [R]
STOP  B STOP
        AREA  INPUT, DATA, READWRITE
COUNT   DCD  0X0
ARRAY16 DCD  0X0
        END .
```

output :

```
R0   0X100000000
R1   0X1000 00014
R2   0X000 00007
R4   0X 0000 0001
```

In memory :

```
0X10000000 :  00000004
              00000001
              00000002
              00000003
              00000001
              00000000 #
```

② Write a program to add two 64 bit numbers.

Program.

```
        AREA  ADD64, CODE, READONLY
        EXPORT  __main
__main
Value 1   DCD   0X11111111
value 2   DCD   0X 22222222
value 3   DCD   0X 33333333
value 4   DCD   0 X 44444444

        LDR    R0, V1
        LDR    R1, V2
        LDR    R2, V3
        LDR    R3, V4
        ADDS   R4, R0, R2
        ADDC   R5, R1, R3
        LDR    R6, = Lower
        STR    R4, [R6]
        LDR    R7, = Higher
        STR    R5, [R7]
~~STOP B STOP~~
        AREA  INPUT, DATA, READWRITE.
HIGHER  DCD  0X0
LOWER   DCD  0X0.

        END.
```

Output :

```
R0  0X11111111
R1  0X22222222
R2  0X33333333
R3  0X44444444
R4  0X44444444
R5  0X66666666
R6  0X10000000
R7  0X10000004
```

(8) Write a program to find square of numbers 1 to 10 using lookup table.

```
    AREA lookup, CODE, READONLY
    EXPORT __main
__main

    LDR   R0, = TABLE1
    LDR   R1, =8
    MOV   R1, R1, LSL #0X2
    ADD   R0, R0, R1
    LDR   R3, [R0]

    NOP
    NOP.

TABLE1      DCD    0X00000000
            DCD    0X00000001
            DCD    0X00000004
            DCD    0X00000009
            DCD    0X00000010
            DCD    0X00000019
            DCD    0X00000024
            DCD    0X00000031
            DCD    0X00000040
            DCD    0X00000051
            DCD    0X00000064.

    END.
```

output :

```
R3  0X00000040
R1  0X00000138
R2  0X00000040.
```

(4) Write an ALP to find largest number in a given array.

```
        AREA LARGE, CODE, READONLY
        EXPORT __main

__main
        LDR  R0, = ARRAY1
        LDR  R1,=5
        LDR  R3, [R0], #4

LOOP
        LDR  R4, [R0], #4
        CMP  R3, R4
        BHI  LARGER

LARGER  SUBS R1, #1
        BNE  LOOP
        LDR  R2, =LARGERNO.
        STR  R3, [R2]
        NOP
        NOP
        AREA INPUT, CODE, READWRITE.

LARGENO    DCD  0X0
ARRAY1     DCD  5.
           DCD  3
           DCD  8
           DCD  10
           DCD  2
        END.
```

O/p :

VNR VJIET

Name of the Laboratory: ____________

Name of the Experiment: ____________

Experiment No: ______ Date: ______

(5) Write a program to find smallest number in a given array.

```
        AREA  SMALL, CODE, READONLY
        EXPORT  __main
__main
        LDR   R0, = ARRAY
        LDR   R1, = 4
        LDR   R3, [R0], #4

LOOP

        LDR   R4 [R0], #4
        CMP   R3, R4
        BLS   SMALLER
SMALLER     SUBS  R1, #1
            BNE  LOOP
            LDR  R2 = SmallNo
            STR  R3, [R2]
            NOP
            NOP
            AREA  INPUT, CODE, READWRITE
SMALL NO    DCD  0X0.
ARRAY       DCD  5
            DCD  3
            DCD  8
            DCD  10
            DCD  2
        END.
```

[illegible]