URBAN EDGE

Lab-01 - Introduction to KEIL uVision-4 And programs on DATA TRANSFER INSTRUCTIONS

1) Write an ARM assembly language problems to store data into general purpose registers

```
AREA RESET, DATA, READONLY
EXPORT __Vectors
__Vectors
    DCD 0X1000100
    DCD Reset_Handler
    ALIGN
    AREA mycode, CODE, READONLY
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    MOV R0, #10
    MOV R1, # 3
STOP
    STOP
    END
```

Output :

R0 0X0000000A

R1 0x00000003

2) Write an ARM assembly language ~~problem~~ program to transfer a 32-bit number from one location in the data memory to another location in the data memory.

```
AREA RESET, DATA, READONLY
EXPORT __Vectors
__Vectors
  DCD 0X10001000
  DCD Reset_Handler
  ALIGN
  AREA mycode, CODE, READONLY
ENTRY
  EXPORT Reset_Handler
Reset_Handler
  LDR R0, =SRC
  LDR R1, = DST
  LDR R3, [R0]
  STR R3, [R1]
STOP B STOP
  AREA mydata, DATA, READWRITE
SRC DCD 0
DST DCD 0
  END
```

Output:

Register	Value	Execution
R0	0X00000048	same
R1	0X10000000	Same
R3	0X00000008	same
SRC	0X00000008	Same
DST	0X00000000	0X00000008

3) Write an ARM assembly language program to transfer block of ten 32 bit numbers from code memory to data memory when the source and destination blocks are nonoverlapping

```
    AREA RESET, DATA, READONLY
    EXPORT __Vectors
__Vectors
    DCD 0X10001000
    DCD Reset_Handler
    ALIGN
    AREA mycode, CODE, READONLY
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    LDR R0, =SRC
    LDR R1, = DST
    MOV R2, #10
LOOP LDR R3, [R0], #4
    STR R3, [R1], #4
    SUBS R2, #1
    BNE LOOP
STOP B STOP
SRC DCD 1,2,3,4,5,6,7,8,9,10
    AREA mydata, DATA, READWRITE
DST DCD 0
    END
```

Output:

R0 0X00000014

R1 0X00000021

R2 0X00000001 → 0X00000002 ... 0X0000000A

SRC 0X00000014 → 0X00000015 ... 0X00000024

DST 0X00000021 → 0X00000022 ... 0X00000023)

URBAN EDGE

④ Reverse an array of 10 32-bit no: in the memory

```
AREA RESET, DATA, READONLY
EXPORT __Vectors
__Vectors
DCD 0x10001000
DCD Reset_Handler
ALIGN
AREA mycode, CODE, READONLY
ENTRY
EXPORT Reset_Handler
Reset_Handler
LDR R0, =SRC
LDR R1, =SRC
ADD R1, #36
LDR R5, =DST
LDR R6, =DST
ADD R6, #36
MOV R2, #5
Loop LDR R3, [R0], #4
LDR R4, [R1], #4
STR R3, [R6], #-4
STR R4, [R5], #4
SUBS R2, R2, #1
BNE Loop
STOP B STOP
SRC DCD 1,2,3,4,5,6,7,8,9,10
AREA mydata, DATA, READWRITE
DST DCD 0
END
```

Output

0x10001000 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A

0x0A 0x09 0x08 0x07 0x06 0x05 0x04 0x03 0x02 0x01