Date :

Exp. No. 01 . a

Exp. Title

Page No. 01.

Write an ALP to find addition of two 32-bit numbers.

```
AREA ADDITION, CODE, READONLY
    MOV R0, # 0x000000ff
    MOV R1, # 0x000000cd
    ADD R2, R0, R1
    END
```

output :-

Before Execution

R0 = 0x00000000
R1 = 0x00000000
R2 = 0x00000000

After Execution

R0 = 0x000000ff
R1 = 0x000000cd
R2 = 0x000001cc

Date :

Exp. No. 01.b

Exp. Title

Page No. 02

Write an ALP to find Substraction value of two 32-bit numbers.

```
AREA SUBSTRACTION, CODE, READONLY.
    MOV R1, #005
    MOV R2, #005
    SUB R3, R1, R0 ; SUB
    END
```

Output :-

Before Execution

R0 = 0x00000000

R1 = 0x00000000

R2 = 0x00000000

After Execution

R0 = 0x00000005

R1 = 0x00000005

R2 = 0x00000000

Output :-

Before Execution	After Execution
R_0 = 0x00000000	R_0 = 0x00000005
R_1 = 0x00000000	R_1 = 0x00000005
R_2 = 0x00000000	R_2 = 0x00000019.

Date :

Exp. No. 01. C

Exp. Title

Page No. 03

Write an ALP to find the multiplication value of two 32-bit number.

```
AREA MUL, CODE, READONLY
     MOV R1, # 0005
     MOV R0, # 0005
     MUL R2, R0, R1
     END
```

Output :-

Before Execution

R0 = 0x00000000

After Execution

R0 = 0x00000037

Memory Allocation

0x40000000

Date :

Exp. No. 02.

Exp. Title

Page No. 04

Write an ALP to find the sum of first 10 integers numbers.

```
AREA ADDITION, CODE, READONLY
START
        MOV R5, #10
        MOV R0, #0
        MOV R1, #1
LOOP    ADD R0, R0, R1
        ADD R1, R1, #1
        SUB R5, R5, #1
        Cmp R5, #0
        BNE LOOP
        LDR R4, = RESULT
        STR R0, [R4]
XSS     B XSS
AREA DATA1, DATA, READWRITE
RESULT  DCD 0X0
        END
```

Date :

Exp. No. 03.

Exp. Title

Page No. 05.

Write an ALP to find factorial of number.

```
AREA FACTORIAL , CODE, READONLY
START
        MOV  R0, #7
        MOV  R1, R0
FACT    SUB  R1, R1, #1
        CMP  R1, #1
        BEQ  STOP
        MUL  R3, R0, R1
        MOV  R0, R3
        BNE  FACT
STOP    NOP
        NOP
        NOP
        END.
```

Output :-

Before Execution

R0 = 0x00000000
R1 = 0x00000000
R2 = 0x00000000
R3 = 0x00000000

After Execution

R0 = 0x000013B0
R1 = 0x00000001
R2 = 0x00000000
R3 = 0x000013B0.

Output :-

Before Execution.

R0 = 0x0000 0000

After Execution.

R0 = 0x00007777
R1 = 0x0000003E
R2 = 0x00004444
R3 = 0x0000ffff
R4 = 0x40000000

Memory Allocation.

0x40000000

Date :	Exp. Title	Page No.
Exp. No. 04.		06.

Write an ALP to add an array of 16-bit numbers and store the 32-bit result in internal RAM.

```
AREA    ADDITION , CODE , READONLY
START
        MOV  R5, #5
        MOV  R0, #0
        LDR  R1 =VALUE1
loop    LDR  R2,[R1], #2
        LDR  R3, MASK
        AND  R2, R2, R3
        ADD  R0, R0, R2
        SUBS R5, R5, #1
        CMP  R5, #0
        BNE  loop
        LDR  R4, =RESULT
        STR  R0, [R4]
XSS     B XSS
MASK    DCD 0X0000FFFF
        DCW 0X1111, 0X2222, 0X4444
AREA DATA2, DATA, READONLY WRITE
RESULT  DCD 0X0
        END
```

Output :-

Before Execution.

R0 = 0x00000000

After Execution

R0 = 0x00000000
R1 = 0x00000037
R2 = 0x00000054
R3 = 0x00000059
R4 = 0x00000000
R5 = 0x40000030

Memory Address

0x40000000 : 00 00 00 00 01 00 00 00 01 00 00
00 02 00 00 00 03 00 00 00 05 00
00 00 08 00 00 00 00 00 00 00 05
00

0x40000022 : 00 00 22 00 00 00 37 00 00 00 59 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00

Date : Exp. No. 05.	Exp. Title	Page No. 07.

Write an ALP to Integer generate the fibonacci Series upto the given number n.

```
AREA FIBN, CODE, READONLY
     MOV R0, #10;
     MOV R1, #0;
     MOV R2, #1;
     LDR R5, = RESULT - SERIES
     STR R1, [R5], #4
     STR R2, [R5], #4
LOOP ADD R3, R1, R2
     STR R3, [R5], #4
     MOV R1, R2
     MOV R2, R3
     SUBS R0, R0, #1
     BNE LOOP
DONE B DONE
 AREA DATA1, DATA, READONLY/WRITE
RESULT - SERIES DCW 0x0000
     END.
```

Output :-

Before Execution

$R_0 = 0x00000000$

After Execution.

$R_0 = 0x00000024$

$R_1 = 0x0000000C$

$R_3 = 0x00000009$

Memory allocation :-

Date :	Exp. Title	Page No.
Exp. No. 06.		08.

Write an ALP to find the square of a numbers (1 to 10) using look-up table

```
AREA    SQUARE, CODE, READONLY
        LDR R0, = TABLE1
        LDR R1, = 7
        MOV R1, R1, LSL #0X2
        ADD R0, R0, R1
        LDR R3, [R0]
XSS     B XSS
TABLE1  DCD 0X00000000
        DCD 0X00000001
        DCD 0X00000004
        DCD 0X00000009
        DCD 0X00000010
        DCD 0X00000019
        DCD 0X00000024
        DCD 0X00000031
        DCD 0X00000040
        DCD 0X00000051
        DCD 0X00000064;
AREA DATA1, DATA, READWRITE
RESULT DCD 0X00000000
        END.
```

INDEX

Sl.No.	Date	Experiment	P. No.	Conduction & Result	Journal Submission	Sign
01.a	06/02/25	Write an ALP to ADD of 2 no's	1			
01b.	06/02/25	Write an ALP to SUB of 2 no's	2			
01.c	06/02/25	Write an ALP to MUL of 2 no's	3			
02.	27/02/25	Write an ALP to find sum of 10 integer numbers	4			
03.		Write an ALP to find factorial of number	5			
04.	23/03/25	Write an ALP to add an array of 16-bit num & store 32 bit result in internal RAM.	6			
05.	06/03/25	Write an ALP to generate the fibonacci series up to given number 'n'	7			
06.	27/03/25	Write an ALP to find the square of a number (1 to 10) using look-up table.	8			