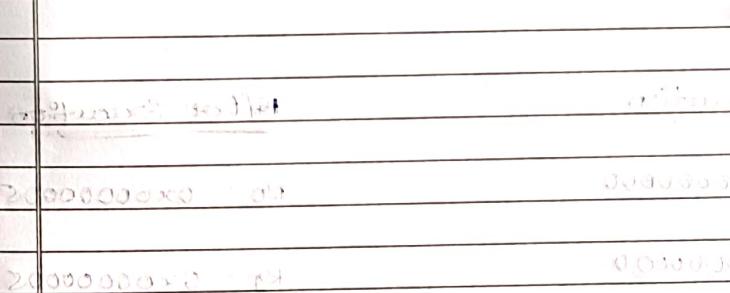
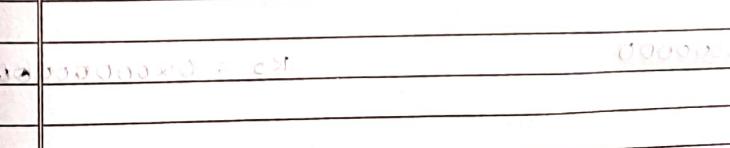


Date :	Exp. Title	Page No.
Exp. No.	01 . a	01.
	Write an ALP to find addition of two 32-bit numbers.	
	AREA ADDITION, CODE, READONLY	
	<pre> MOV R0, #0x000000ff MOV R1, #0x000000cd ADD R2, R0, R1 END </pre>	
		
		

1. **प्राचीन विद्या**
2. **विद्यालय**
3. **विद्यार्थी**
4. **विद्यालयीकरण**
5. **विद्यालयीकरण के लाभ**

Output:

Before Execution

$R_0 = 0200000000$

R1 = 0x00000000

$R_2 = \text{DEBDBDDDD}$

R₂ = 0x00000000

After Execution

$$R0 = 0x00000005$$

$$R_1 = 0x00000005$$

$$R_2 = 0x00000000$$

Date : Exp. No. 01.b	Exp. Title	Page No. 02
	Write an ALP to find Subtraction value of two 32-bit numbers.	
	AREA SUBTRACTION, CODE, READONLY. MOV R1, #005 MOV R2, #005 SUB R3, R1, R0 ; SUB END	
		Ans: 11111111 00000000 00000000 00000000
Q1) 10000000 00000000	ans1) 11111111 00000000 00000000 00000000	
Q2) 00000000 11111111	ans2) 11111111 00000000 00000000 00000000	
Q3) 11111111 00000000	ans3) 11111111 00000000 00000000 00000000	

ଅନୁଷ୍ଠାନିକ ପରୀକ୍ଷା ମଧ୍ୟ ପରିଚ୍ଛନ୍ନ ହେଲାମୁଣ୍ଡିଲା
ଏହାରେ ପରିଚ୍ଛନ୍ନ ହେଲାମୁଣ୍ଡିଲା
ଏହାରେ ପରିଚ୍ଛନ୍ନ ହେଲାମୁଣ୍ଡିଲା

Output :-

Before execution

$R_0 = 0x000000000$

R1 := 0x000000000

$$R_2 = 0.2000000000$$

After Execution

$$R_0 = 0x000000005$$

R1 = 0x00000005

$$R_2 = 0x00000019.$$

Date :	Exp. Title	Page No.
Exp. No. 01. C		03
Write an ALP to find the multiplication value of two 32-bit number.		
<pre> AREA mul, CODE, READONLY MOV R1, #0005 MOV R0, #0005 MUL R2, R0, R1 END </pre>		
<p>Inputs:</p> <p>R1: 00000005 R0: 00000005</p>		Outputs: R2: 0000000000000025 = 01
<p>Inputs:</p> <p>R1: 00000005 R0: 00000005</p>		Outputs: R2: 0000000000000025 = 01

DATA SEGMENT
VAR1 DB 0AH
VAR2 DB 0BCH
VAR3 DB 0CCH

MOV AX, VAR1
MOV BX, VAR2
MOV CX, VAR3
ADD AX, BX
ADD AX, CX
MOV DS, AX
JMP DS:
PRINT

Output:

Before Execution

RO = 0x00000000

Memory Allocation

0x40000000

After Execution

RO = 0x00000037

Date :	Exp. Title	Page No.
Exp. No. 02.	Write an ALP to find the sum of first 10 integers numbers.	04
	AREA ADDITION, CODE, READONLY	
	START	
	MOV R5, #10	
	MOV RO, #0	
	MOV R1, #1	
	LOOP ADD RO, RO, R1	
	ADD R1, R1, #1	
	SUB RS, RS, #1	
	CMP RS, #0	
	BNE LOOP	
	LDR R4, =RESULT	
	STR RO, [R4]	
	XSS B XSS	
	AREA DATA 1, DATA, READWRITE	
	RESULT DCD 0X0	
	END	

DATA SEGMENT

VAR1 DB 0A

VAR2 DB 0B

VAR3 DB 0C

VAR4 DB 0D

DATA SEGMENT

RESULT DB 00

DATA1 DB 00

DATA2 DB 00

DATA3 DB 00



Scanned with OKEN Scanner

Date : 08.09.2019
 Exp. No. 03.
 Title : Write an ALP to find factorial of number.
 Page No. 05.

AREA FACTORIAL, CODE, READONLY
 START
 MOV R0, #1
 MOV R1, R0
 FACT SQB 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 = 0x00000000

Date :	Exp. Title	Page No.
Exp. No. 03.	Write an ALP to find factorial of number.	05.
AREA FACTORIAL, CODE, READONLY		
START		
MOV R0, #1		
MOV R1, R0		
FACT SQB R1, R1, #1		
CMP R1, #1		
BEQ STOP		
MUL R3, R0, R1		
MOV R0, R3		
BNE FACT		
STOP NOP		
NOP		
NOP		
END.		



Question :- Develop a program to add an array of 16-bit numbers and store the 32-bit result in Internal RAM.

```

    AREA DATA, DATA, READONLY
    START
        LDR R0, #00000000
        LDR R1, #00000000
        LDR R2, #00000000
        LDR R3, #00000000
        LDR R4, #00000000
        ADD R0, R1, R2
        ADD R0, R0, R3
        ADD R0, R0, R4
        STR R0, [R0]
        ADD R1, R1, R2
        ADD R1, R1, R3
        ADD R1, R1, R4
        STR R1, [R1]
        ADD R2, R2, R3
        ADD R2, R2, R4
        STR R2, [R2]
        ADD R3, R3, R4
        STR R3, [R3]
        ADD R4, R4, R4
        STR R4, [R4]
    END

```

Output :-

Before Execution.

R0 = 0x0000 0000

After Execution.

R0 = 0x0000 0fff

R1 = 0x0000 003E

R2 = 0x0000 4444

R3 = 0x0000 ffff

R4 = 0x4000 0000

Memory Allocation.

0x4000 0000

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
        LDR R0, #00000000
        MOV R5, #5
        LDR R0, [R0]
        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 WRTWE
    RESULT DCD 0x00000000
    END

```



Exp. No. 05.

07.

Output: The output of the program is given below in the form of memory dump and register dump.

Before Execution:

R0 = 0x00000000

After Execution:

R0 = 0x00000000

R1 = 0x00000037

R2 = 0x00000054

R3 = 0x00000059

R4 = 0x00000000

R5 = 0x40000030

R6 = 0x00000000

R7 = 0x00000000

R8 = 0x00000000

R9 = 0x00000000

Memory Address

0x40000000 : 00 00 00 00 01 00 00 00 01 00 00 00

00 02 00 00 00 03 00 00 00 05 00 00 00

00 00 08 00 00 06 00 00 00 00 00 05

00 10000000 00000000

0x40000022 : 00 00 220000 000000 37000000 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 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

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, READON/WRITE

RESULT - SERIES DCW 0x0000

END.



Output: A program to print the square of numbers from 1 to 10 using look-up table.

Before Execution

R0 = 0x00000000

After Execution.

R0 = 0x00000024

R1 = 0x0000000C

R3 = 0x00000009

PC = 0x00000000

LR = 0x00000000

Memory Allocation:

TABLE1 : 0x00000000 - 0x0000000F

99A 99B 99C

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

14 14 14

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

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

AREA SQUARE, CODE, READONLY

LDR R0, = TABLE1

LDR R1, = ?

MOV R1, R1, LSL #0x22

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