

I. Write programs in ARM assembly language to perform 32-bit addition and subtraction using:

a. direct addressing mode

b. indirect addressing mode

b. barrel shifter

Program Ia: Direct addressing mode

Addition:

```
                area program,code,readonly
entry
main
    LDR R1,value1
    LDR R2,value2
    ADD R3,R1,R2
    SWI &11

                area program,data,readonly
value1 DCD &00000005
value2 DCD &00000030

    END
```

Subtraction:

```
                area program,code,readonly
entry
main
    LDR R1,value1
    LDR R2,value2
    SUB R3,R1,R2
    SWI &11

                area program,data,readonly
value1 DCD &00000005
value2 DCD &00000003

    END
```

Program 1b: Indirect addressing mode

Addition:

```
                area program,code,readonly
entry
main
    LDR R0,value1
    LDR R1,value2
    LDR R2,[R0]
    LDR R3,[R1]
    ADD R4,R2,R3
    SWI &11
                area program,data,readonly
value1 DCD 0x00003000
value2 DCD 0x00003003
    END
```

Subtraction:

```
                area program,code,readonly
entry
main
    LDR R0,value1
    LDR R1,value2
    LDR R2,[R0]
    LDR R3,[R1]
    SUB R4,R2,R3
    SWI &11
                area program,data,readonly
value1 DCD 0x00003006
value2 DCD 0x00003009
    END
```

Program 1c: Barrel Shifter

Addition:

```
        area program,code,readonly
entry
main
    LDR R1,value
    MOV R2,R1,LSL#0x02
    ADD R3,R1,R2
    SWI &11
        area program,data,readonly
value DCD &00000003
    END
```

Subtraction:

```
        area program,code,readonly
entry
main
    LDR R1,value
    MOV R2,R1,LSL#0x04
    ADD R3,R2,R1
    SWI &11
        area program,data,readonly
value DCD &00000005
    END
```

II. Write a program in ARM assembly language to perform left and right shifts on a number.

Program II

```
        area program,code,readonly
entry
main
    LDR R1,value
    MOV R2,R1,LSL#0x02
    MOV R3,R1,LSR#0x05
    MOV R4,R1,ASR#0x04
    MOV R5,R1,ROR#0x03
    SWI &11
        area program,data,readonly
value DCD &00000003
END
```

III. Write a program in ARM assembly language to compute one's complement of a number.

Program III

```
        area program,code,readonly
entry
main
        LDR R1,value
        MVN R1,R1
        SWI &11
        area program,data,readonly
value DCD &00000043
        END
```

IV. Write a program in ARM assembly language to find whether a number is even or odd.

Program IV

```
        area program,code,readonly
entry
main
        LDR R1,value
        MOV R2,#0x01
        AND R3,R1,R2
        SWI &11
        area program,data,readonly
value DCD &00000043
        END
```

V. Write a program in ARM assembly language to perform multiplication using addition.

Program V

```
                area program,code,readonly
entry
main
    LDR R0,value1
    LDR R1,value2
    MOV R2,R0
    MOV R3,#0x01
LOOP
    ADD R3,R3,#0x01
    ADD R0,R0,R2
    CMP R1,R3
    BNE LOOP
    SWI &11
                area program,data,readonly
value1 DCD &00000002
value2 DCD &00000006
    END
```

NOTE: How does CMP, R1,R2 differ from SUB R1,R2?

CMP updates a flag, which BNE then checks.

VI. Write a program in ARM assembly language to store multiplication table of a number.

Program VI

```
                area program,code,readonly
entry
main
    LDR R0,value1
    LDR R1,value2
    MOV R2,#0x0A
    MOV R3,R0
LOOP
    STR R0,[R1]
    ADD R0,R0,R3
    SUB R2,R2,#0x01
    ADD R1,R1,#0x04
    CMP R2,#0x00
    BNE LOOP
    SWI &11
                area program,data,readonly
value1 DCD &00000003
value2 DCD &00000080
    END
```

NOTE: Multiplication table of 3 in hexadecimal?

VII. Write a program in ARM assembly language to perform division using subtraction.

Program VII

```
                area program,code,readonly
entry
main
    LDR R0,dividend
    LDR R1,divisor
    MOV R2,#0x00
    MOV R3,R0
LOOP
    SUB R3,R3,R1
    ADD R2,R2,#0x01
    CMP R3,R1
    BGE LOOP
    SWI &11
                area program,data,readonly
dividend DCD &0000000A
divisor DCD &00000002
    END
```

VIII. Write a program in ARM assembly language to count the number of characters in a string.

Program VIII

```
        area program,code,readonly
entry
main
        LDR R0,=string
        MOV R2,#0x00
LOOP
        LDRB R1,[R0],#0x01
        CMP R1,#0x00
        ADDNE R2,R2,#0x01
        BNE LOOP
        SWI &11
        area program,data,readonly
string DCB "ABCDEF"
        END
```

IX. Write a program in ARM assembly language to count the number of occurrences of a particular character in a string.

Program IX

```
        area program,code,readonly
entry
main
        LDR R0,string
        MOV R2,#0x00
LOOP
        LDRB R1,[R0],#0x01
        CMP R1,#"S"
        ADDEQ R2,R2,#0x01
        CMP R1,#0x00
        BNE LOOP
        SWI &11
        area program,data,readonly
string DCB "MISSISSIPPI"
        END
```

X. Write a program in ARM assembly language to add two integer strings.

Program X

```
                area program,code,readonly

entry
main
    LDR R0,=val1
    LDR R1,=val2
    LDR R2,=val3
    LDR R3,count
LOOP
    LDRB R4,[R0],#0x01
    LDRB R5,[R1],#0x01
    ADD R6,R4,R5
    STRB R6,[R2],#0x01
    SUB R3,#0x01
    CMP R3,#0x00
    BNE LOOP
    SWI &11

                area program,data,readonly
count DCD &00000004
val1 DCB 1,2,3,4
val2 DCB 5,6,7,8
val3 DCD &00000000

                END
```

XI. Write a program in ARM assembly language to find the factorial of a number.

Program XI

```
                area program,code,readonly

entry
main
    LDR R0,value1

    MOV R1,#0x01

LOOP
    MUL R2,R1,R0

    MOV R1,R2

    SUB R0,R0,#0x01

    CMP R0,#0x01

    BGT LOOP

                area program,data,readonly

value1 DCD &00000004

    END
```

XII. Write a program in ARM assembly language to perform addition of two 64-bit numbers.

Program XII

```
        area program,code,readonly

entry
main
    LDR R0,=value1

    LDR R1,[R0]

    LDR R2,[R0,#0x04]

    LDR R0,=value2

    LDR R3,[R0]

    LDR R4,[R0,#0x04]

    ADDS R5,R2,R4

    ADC R6,R1,R3

    LDR R1,=result

    STR R6,result

    STR R5,[R1,#0x04]

        area program,data,readonly

value1 DCD &12A2E640,&F2100123

value2 DCD &001019BF,&40023F51

result DCD &00000000

END
```

XIII. Write a program in ARM assembly language to find the largest number in an array.

Program XIII

```
                area program,code,readonly

entry
main
    LDR R0,=val1
    LDRB R1,[R0]
    LDR R2,count
LOOP
    LDRB R3,[R0],#0x01
    CMP R3,R1
    MOVGT R1,R3
    SUB R2,#0x01
    CMP R2,#0x00
    BNE LOOP
    SWI &11

                area program,data,readonly
count DCD &00000006
val1 DCB 1,2,4,7,5,6

    END
```

XIV. Write a program in ARM assembly language to copy an array.

Program XIV

```
        area program,code,readonly

entry
main
        LDR R0,array

        MOV R4,#0x04

        LDR R5,value

LOOP
        LDRB R2,[R0],#0x01

        STRB R2,[R5],#0x01

        SUB R4,R4,#0x01

        CMP R4,#0x00

        BNE LOOP

        SWI &11

        area program,data,readonly

array DCB 1,2,3,4

value DCD &10000080

        END
```


XV. Write a program in ARM assembly language to implement the following equations:

a. ax^2+by^2

b. $6(x+y)+2z+4$

Program XVa

```
        area program,code,readonly

entry
main
    LDR R0,value1

    LDR R1,value2

    LDR R2,value3

    LDR R3,value4

    MUL R4,R2,R2

    MUL R5,R4,R0

    MUL R6,R3,R3

    MUL R7,R6,R1

    ADD R8,R5,R7

        area program,data,readonly

value1 DCD &00000001

value2 DCD &00000002

value3 DCD &00000003

value4 DCD &00000004

    END
```

NOTE: $a = 1, b = 2, x = 3, y = 4$

Program XVb

```
                area program,code,readonly

entry

main

    LDR R0,value1

    LDR R1,value2

    LDR R2,value3

    ADD R3,R0,R1

    MUL R3, #0x06

    MUL R2, #0x02

    ADD R5,R3,R2,#0x04

                area program,data,readonly

value1 DCD &00000001

value2 DCD &00000002

value3 DCD &00000003

    END
```

NOTE: $x = 1$, $y = 2$, $z = 3$