**1) WAP an assembly language program to perform arithmetic operation using 8051**

ADD

```
MOV A, #25H     ; First number
ADD A, #15H     ; Add second number
MOV R0, A       ; Store result
END
```

SUB

```
MOV A, #45H     ; First number
SUBB A, #20H    ; Subtract second number
MOV R0, A
END
```

MUL

```
MOV A, #12H
MOV B, #05H
MUL AB          ; A = low byte, B = high byte
MOV R0, A
MOV R1, B
END
```

DIV

```
MOV A, #64H     ; Dividend
MOV B, #08H     ; Divisor
DIV AB          ; A = quotient, B = remainder
MOV R0, A
MOV R1, B
END
```

**2.write assembly language prg to perform logical operation using 8051 microcontroller**

AND

```
MOV A, #55H
ANL A, #0F0H
MOV R0, A
END
```

OR

```
MOV A, #22H
ORL A, #11H
MOV R0, A
END
```

NOT

```
MOV A, #55H
CPL A
MOV R0, A
END
```

XOR

```
MOV A, #5AH
XRL A, #3CH
MOV R0, A
END
```

**3).Write an assembly Language Program to transfer a block of n byte of data from the source (20H) to the destination (40h) using Internal RAM**

```
ORG  0000H
MOV R0,#20H
MOV R1,#40H
MOV R2,#05H
HERE: MOV A,@R0
MOV @R1,A
INC R0
INC R1
DJNZ R2, HERE
XRL A,B
MOV R0,A
END
```

4.Write an assembly language program to execute source work starting with address 20h (internal ram) containing n(0.5)

```
ORG 0000H
MOV R0,#20H
MOV R1,#40H
MOV R3,#05H
HERE: MOV A,@R0
XCH A,@R1
MOV @R0,A
INC R0
INC R1
DJNZ R2, HERE
END
```

5.

```
ORG 0000H
MOV R0,#34H
MOV A,@R0
INC R0
SUBB A,@R0
MOV R5,A
JC DOWN
MOV R6,#01H
SJMP DONE
DOWN:
MOV R6,#00H
DONE: NOP
END
```

6.

```
ORG 0000H
MOV R0,#34H
MOV A,@R0
INC R0
SUBB A,@R0
MOV R5,A
JC DOWN
MOV R6,#01H
SJMP DONE
DOWN:
MOV R6,#00H
DONE: NOP
END
```

7.

```
ORG 0000H

MOV A,30H

MOV A,31H

MUL AB

MOV 32H,A

MOV 35H,B

END
```

8.

```
ORG 0000H

MOV A,30H

MOV A,31H

DIV AB

MOV 32H,A

MOV 33H,B

END
```

9.1

```
ORG 0000H

MOV R0,#20H

MOV R1,#40H

MOV R2,#05H

BACK:

MOV A,@R0

JB ACC.7,SKIP

MOV @R1,A

INC R1

SKIP:

INC R0

DJNZ R2,BACK
```

```
        END
```

9.2

```
        ORG 0000H
        MOV R0,#20H
        MOV R1,#40H
        MOV R2,#05H
        BACK:
        MOV A,@R0
        JNB ACC.7,SKIP
        MOV @R1,A
        INC R1
        SKIP:
        INC R0
        DJNZ R2,BACK
        END
```

10.1-2

```
        ORG 0000H
        MOV R0,#20H
        MOV R1,#40H
        MOV R2,#05H
        BACK:
        MOV A,@R0
        JNB ACC.0,SKIP
        MOV @R1,A
        INC R1
        SKIP:
        INC R0
        DJNZ R2,BACK
        END
```