

EXP2

Code [HARDWARE]:

```
MOV AX,0000H
MOV DS, AX
MOV SI,2000H
XOR AX, AX
MOV CL,05H
ADD AX, [SI]
INC SI
LOOP 100F
MOV [3000], AX
HLT
```

EXP 3

Code:

```
MOV AX, 0000H
MOV DS, AX
MOV CH, 05H
MOV CL, 05H
MOV SI,2000H
MOV AL, [SI]
MOV BL, [SI+1]
CMP AL, BL
JLE 1021
MOV DL, [SI+1]
XCHG [SI], DL
MOV [SI+1], DL
INC SI
DEC CL
JNZ 1010
DEC CH
JNZ 1009
HLT
```

EXP 4

1. BCD to HEX Conversion:

Code:

```
MOV SI, 2000H
MOV DI, 3000H
MOV BL, [SI]
AND BL, 0F
MOV AL, [SI]
AND AL, 0F0
MOV CL, 04
ROR AL, CL
MOV DL, 0A
MUL DL
ADD AL, BL
MOV [DI], AL
HLT
```

2. BCD to ACSII Conversion

Code:

```
MOV AL, [2000]
MOV AH, AL
AND AL, 0F
```

```
MOV CL, 04
SHR AH, CL
OR AX, 3030
MOV [3000], AX
HLT
```

EXP 5

1. Factorial of a number

```
MOV CX, [2000]
MOV AX, 0001
MUL CX
DEC CX
JNZ 1008
MOV [3000], AX
HLT
```

3. Combination

```
MOV DX, [2000]
MOV CX, [2002]
SUB DX, CX
MOV CX, DX
MOV AX, 0001
MUL CX
DEC CX
JNZ 1010
MOV BX, AX
MOV CX, [2002]
MOV AX, 0001
MUL CX
DEC CX
JNZ 101F
MUL BX
MOV BX, AX
MOV CX, [2000]
MOV AX, 0001
MUL CX
DEC CX
JNZ 1030
DIV BX
MOV [3000], AX
HLT
```

To find sum without carry:

```
ORG 0000H
MOV A,#21H
ADD A,#02H
MOV 03H,A
END
```

To find sum with carry:

```
ORG 0000H
SETB C
MOV A,#21H
ADD A,#42H
MOV 03H,A
END
```

EXP 7

Subtraction:

```
ORG 0000H
CLR C
MOV A,#5FH
SUBB A,#8AH
JNC NEXT
CPL A
INC A
NEXT: MOV R1,A
END
```

Subtraction:

```
ORG 0000H
CLR C
MOV A,#56H
SUBB A,#78H
MOV R7,A
MOV A,#27H
SUBB A,#12H
MOV R6,A
END
```

Multiplication:

```
ORG 0000H
MOV A,#30H
MOV B,#69H
MUL AB
END
```

Division:

```
ORG 0000H
MOV A,#95
MOV B,#05
DIV AB
END
```

1. Write a program to clear accumulator [A] then add 5, 5 times to the accumulator
2. Write an 8051 ASM program to read a temperature value (T) from RAM onto the location 55H. According to the test results place the temperature value into the register located below
3. Write a program to move a block of 5 data starting from RAM memory 40H to external ROM memory 3000H onwards and perform complement operation before storing into the ROM.

PROGRAM 1:

```
ORG 0000H
MOV A, #00
MOV R0, #5
LOOP: ADD A, #5
DJNZ R0, LOOP
MOV R5, A
END
```

PROGRAM 2:

```
ORG 0000H
MOV A, 55H
CJNE A, #50, OVER
SJMP EXIT
OVER: JNC NEXT
MOV R1, A
SJMP EXIT
NEXT: MOV R2, A
EXIT: NOP
END
```

PROGRAM 3:

```
ORG 0000H
MOV R0, #40H
MOV DPTR, #3000H
MOV R2, #05H
LOOP: MOV A, @R0
CPL A
MOVX @DPTR, A
INC R0
INC DPTR
DJNZ R2, LOOP
END
```

1. Write an 8051 assembly language program to toggle the bits of P1 for every 500ms. Assume the crystal frequency as 11.0592MHz
2. Write an 8051 assembly language program using timers to generate a frequency of 2KHz on pin port P2.7. Assume the crystal frequency as 11.0592MHz
3. Assume that the clock pulses are fed into pin T1(P3.5), Write an 8051 assembly language program for counter 1 in mode 2 to count the pulses and display the state of the TL1 count on P2 which connects to 8 LEDs

PROGRAM 1:

```
DELAY : MOV R2, #04H
HERE3 : MOV R1, #0FFH
HERE2 : MOV R0, #0FFH
HERE1 : DJNZ R0, HERE1
        DJNZ R1, HERE2
        DJNZ R2, HERE3
        RET
```

PROGRAM 2:

```
ORG 0000H
MOV
TMOD, #01H
BACK : MOV
TL0, #01AH
MOV TH0, #0F0H
SETB TR0
AGAIN : JNB
TF0, AGAIN
CLR TR0
CPL P2.7
CLR TF0
SJMP BACK
```

PROGRAM 3:

```
ORG 0000H
REPEAT : MOV TMOD, #20H
MOV TH1, #0
SETB P3.5
AGAIN : SETB TR1
BACK : MOV A, TL1
MOV P1, A
CLR P3.5
SETB P3.5
JNB TF1, BACK
CLR TR1
CLR TF1
SJMP REPEAT
END
```

1. For 8051 to transfer "VIT" serially at 9600 baud, 8-bit data, 1 stop-bit, do this continuously.
2. For 8051 to receive bytes data serially, and put them in P1, set the baud rate at 4800, 8-bit data, 1 start and 1 stop bit.
3. To receive a number (between 1-9) continuously from the serial port and send the square of the received number continuously to P2. Assume the baud rate for serial communication to be 1200 and XTAL = 11.0592

MHz Consider Mode 2

operation for Timer 1.

Task 1.

```
MOV TMOD, #20H ; TIMER1,
MODE 2(AUTO
RELOAD)
MOV TH1, #0FDH ; 9600 BAUD
RATE
MOV SCON, #50H ; 8 BIT, 1 STOP,
REN ENABLED
SETB TR1 ; START TIMER 1
AGAIN: MOV A, #'V'
ACALL TRANS
MOV A, #'I'
ACALL TRANS
MOV A, #'T'
ACALL TRANS
MOV A, #' '
ACALL TRANS
SJMP AGAIN ; KEEP REPEATING
IT
; SERIAL DATA TRANSFER
SUBROUTINE
TRANS: MOV SBUF, A ; LOAD
SBUF
HERE: JNB TI, HERE ; WAIT FOR
THE LAST BIT
CLR TI ; GET READY FOR THE
NEXT BYTE
```

RET

END

Task 2)

```
ORG 0000H
LJMP MAIN
ORG 0100H
MAIN:
MOV TMOD, #20H
MOV TH1, #0FAH
MOV SCON, #50H
SETB TR1
HERE: JNB RI, HERE
MOV A, SBUF
MOV P1, A
CLR RI
CJNE A, #'$', HERE
END
```

Task 3)

```
ORG 0000H
MOV TMOD, #20H
MOV TH1, #0E8H
MOV SCON, #50H
MOV P2, #00H
SETB TR1
RPT: JNB RI, RPT
MOV A, SBUF
CLR C
SUBB A, #30H
MOV B, A
MUL AB
MOV P2, A
CLR RI
SJMP RPT
END
```