Key

Q1

a) CS = 4000H and DS = 4500H. Find if there is any overlap between these two segments. If so, what is the size of the overlap.

Code segment range = 40000H to 4FFFFH and data segment range = 45000H to 54FFFH
Overlap size = 4FFFF – 45000 = AFFFH or 43 KB

b) Find the errors in the following instructions (if so).
i. MOV CS,5000H ---- MOV CS, AX
ii. SBB [AX], [2000H] ---- SBB AX, [2000H]
iii. MUL AL, BL ---- MUL BL
iv. AL =83H; ---- AL = 12, CY = 1
CL=29H;
ADD AL, CL;
DAA;
After execution (AL)=05, carry flag (CY)=0;
v. OR AX, 0098H contents of AX = 3F0FH, after execution AX = 3090H. ----- AX =3F9FH

Q2

Write an 8086 ALP to find (compute) the values of the function $f(x) = x^2 + 3x + 2$ in the range x=1 to x=10 and store them in the memory location from 5000H onwards.

```
ASSUME CS:CODE, DS:DATA

DATA SEGMENT

   NUMLIST DB 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH

   NEXT DB 0FF00H  DUP (00H)  // REMAINING VALUES WILL BE ZEROS

   COUNT EQU 0AH


DATA ENDS




CODE SEGMENT

   START:

   MOV AX, DATA      ; get the segment address of DATA segment in DS register

   MOV DS, AX

   MOV CX, COUNT     ; COUNT is the number of data has to be added
```

```
    MOV SI, OFFSET NUMLIST    ; get the offset address of NUMLIST
    MOV DI, 5000H
    MOV AX, 0000H
    MOV BX, 0000H
LP: MOV BL, [SI]
    MOV AL, BL
    ADD AL, 03H
    MUL BL
    ADD AL, 02H
    MOV [DI], AL
    INC DI
    INC SI
    DEC CX
    JNZ LP
    MOV AH, 00H
    INT 21H
    CODE ENDS
    END START
```

RET

Q3

Write an 8051 ALP to multiply two numbers without using MUL AB instruction and assume that the product does not exceeds 8-bits. Display the digits of the product on the seven segment displays connected to P0 and P1.

```
ORG 0000H
; 7 SEGMENT LOOKUP TABLE
MOV 30H, #03FH
MOV 31H, #06H
MOV 32H, #5BH
MOV 33H, #04FH
```

```
MOV 34H, #066H
MOV 35H, #06DH
MOV 36H, #07DH
MOV 37H, #07FH
MOV 38H, #07FH
MOV 39H, #067H

MOV R2, #04H ; INPUT-1
MOV R3, #0CH ; INPUT-2
MOV A, #00H
L: ADD A, R3
DJNZ R2, L
MOV R2, A

MOV B, #100D
DIV AB
MOV A, B
MOV B, #10D
DIV AB
MOV R6, A
MOV R7, B

MOV A, #30H
ADD A, R6
MOV R0, A
MOV A, @R0
MOV P0, A

MOV A, #30H
ADD A, R7
MOV R0, A
MOV A, @R0
MOV P1, A

S:SJMP S

END
```

Q4

Write an 8051 (AT89C51) ALP to generate a periodic waveform simultaneously as follows: i) 0.1ms ON & 0.1ms OFF ii) 0.2ms ON & 0.2ms OFF using timer. Don't access TL0/1 register in the ALP. Provide necessary timer calculations.

i)
Delay = 0.1ms
Number of count = 0.1ms/1.085us = 92.16 = 92
TH0 = 255 – 92 = 163 or 0A3H
ii)

Delay = 0.2ms
Number of count = 0.2ms/1.085us = 184.32 = 184
TH0 = 255 – 184 = 71 or 47H

```
ORG 0000H
LJMP MAIN

ORG 000BH ;Timer 0 interrupt vector table
CPL P2.1 ;toggle P2.1 pin
RETI

ORG 001BH ;Timer 1 interrupt vector table
CPL P2.2 ;toggle P2.1 pin
RETI

ORG 0050H ;after vector table space
MAIN: MOV TMOD,#22H ;Timer 0, mode 2
MOV TH0,#043H ;TH0=A4H for -92
MOV TH1,#0A3H ;TH0=A4H for -92
MOV IE,#8AH ;IE=10001010 (bin) enable
;Timer 0
SETB TR0 ;Start Timer 0
SETB TR1
BACK: SJMP BACK ;get data from P0
END
```

Q5

Write 8051 (AT89C51) ALP for the following logic. The existing values of TCON=01H and IE = 0FFH.

```
ORG 0000H
LJMP MAIN ;by-pass interrupt
;vector table
;--ISR for INT1 to turn on LED
ORG 0013H ;INT1 ISR
MOV A, #0AH
JA:MOV P1, A
ACALL DELAY
DEC A
CJNE A, #00H, JA

RETI ;return from ISR
;--MAIN program for initialization
ORG 30H
MAIN: MOV IE,#0FFH
MOV TCON, 01H;enable external INT 1
HERE: MOV A, #01H
J:MOV P0, A
ACALL DELAY
INC A
```

```
CJNE A, #0BH, J
SJMP HERE

DELAY:
MOV R5,#50
H3: MOV R4,#255
H2: MOV R3,#255
H1: DJNZ R3,H1
DJNZ R4,H2
DJNZ R5,H3
RET

END
```