LAB PROGRAMS

Assembly language program 8085

1. Add two 8 bit numbers :

Add the contents of memory locations 2000H and 2001H and place the result in memory location 2002 H .

2000 H = 14H, 2001 H = 89 H

Result = 14 H + 89 H = 9DH

Source :

```
LXI  H, 2000H          ; HL Points 2000H
MOV  A, M              ; Get first operand
INX  H                 ; HL points 2001H
ADD M                  ; Add second operand with ACC.
INX  H                 ; HL points to 2002 H
MOV M, A               ; Store the result at 2002 H
RST 0                  ; Stop
```

2. Subtract two 8 bit numbers :

Subtract the contents of memory location 2001H from the memory location 2000H and place the result in memory location 2002 H

2000H = 51H , 2001H = 19 H

Result = 51 – 19 = 38 H

Source :

```
LXI  H, 2000H          ; HL points 2000H
MOV A, M               ; Get first operand
INX  H                 ; HL points 2001 H
SUB  M                 ; Subtract second operand
INX  H                 ; HL points to 2002 H
MOV  M, A              ; Store the result at 2002
HLT                    ; Stop
```

3. Add two 16 bit numbers :

Sample problem :

Memory location 2000H = 15 H, 2002 H = B7 H
2001H= 1 CH 2003 H = 5AH

Result = 1C 15 + 5A B7 = 76 CCH

2004 = CCH

2005 = 76 H

Source 1 :

```
LHLD  2000H      ; Get first 16 bit no in HL
XCHG             ; Save first 16 bit no in DE
LHLD 2002 H      ; Get 2nd 16 bit no in HL
MOV A, E         ;  Get lower byte of the first no
ADD  L           ; Add lower byte of the 2 nd no
MOV L,A          ; Store result in L  register
MOV A, D         ; Get  higher byte of first no
ADC   H          ; Add higher byte of the 2nd no with carry .
MOV  H, A        ; Store result in H register
SHLD 2004 H      ; Store 16 bit result in memory location 2004 H and 2005 H
RST 0            ; Stop
```

Source 2 :

```
LHLD  2000H      ; Get first 16 bit no in HL
XCHG             ; Save first 16 bit no in DE
LHLD 2002 H      ; Get 2nd 16 bit no in HL
DAD  D           ; Add DE and HL
SHLD 2004 H      ; Store 16 bit result in memory location 2004 H and 2005 H
RST 0            ; Stop
```

4. Subtract two 16 bit numbers :

Sample problem :

Memory location 2000H = 19 H, 2002 H = 15 H
2001H= 6AH 2003 H = 5CH

Result = 6A19 - 5C15 = 0E04H
2004 = 04H
2005 = 0E H

Source :

```
LHLD  2000H        ; Get first 16 bit no in HL
XCHG               ; Save first 16 bit no in DE
LHLD 2002 H        ; Get 2nd 16 bit no in HL
MOV A, E           ;  Get lower byte of the first no
SUB  L             ; Subtract  lower byte of the 2 nd no
MOV L,A            ; Store result in L  register
MOV A, D           ; Get  higher byte of first no
SBB   H            ; Subtract  higher byte of the 2nd no with borrow .
MOV  H, A          ; Store result in H register
SHLD 2004 H        ; Store 16 bit result in memory location 2004 H and 2005 H
RST 0              ; Stop
```

5 Bit manipulation :

5.1.Find the 2’s complement of a number

Sample problem

Memory location 2200 H = 55H

Result = 2300H = AAH + 1 = ABH

Source:

```
LDA 2200H    ; Get the number
CMA          ; complement the number
ADD, 01H     ; Add one in the number
STA 2300 H   ; Store the result
HLT          ; Stop
```

5.2. Pack the two unpacked BCD numbers

Sample problem :

2200H = 04

22011H = 09

Result = 2300H = 94

Source :

```
LDA 2201 H   ; Get the most significant BCD digit
RLC
RLC
RLC
RLC          ; Adjust the position
AN1 F0H      ; Make least significant BCD digit zero
MOV C, A     ; Store the partial result
LDA 2200H    ; Get the lower BCD digit
ADD C        ; Add lower BCD digit
STA 2300 H   ; Store the result
HCT          ; Stop
```

5.3.Unpack the BCD number :

Sample problem :

2200 H = 58

Result = 2300 H = 08

2301 H = 05

Source :

```
LDA 2200H     ; Get the packed BCD number
ANI F0 H     ; mask lower nibble
RRC
RRC
RRC
RRC          ; Adjust higher BCD digit as a lower digit
STA 2300 H   ; Store the partial result
LDA 2200 H   ; Get the original BCD number
ANI OF H     ; mask higher nibble
```

```
STA 2201 H    ; Store the result
HLT           ; Stop
```

6.Calculate the sum of series of number:

Calculate the sum of series of numbers. The length of the series is in memory location 2200 H and series itself begins from memory location 2201H

a. Assume the sum to be 8 bit number so you can ignore carries, store the sum at memory location 2300H.

b. Assume the sum to be 16 bit number store the sum at memory location 2300H and 2301H

a. Sample problem

2200 H = 04 H , →counter
2201 H = 20H , 2202 H = 15 H ,
2203 H = 13 H ,2204 H = 224 .

Result = 20 + 15 + 13 + 22 = 6 A H

2300 H = 6 A H

Source :

```
            LDA 2200 H
            MOV C, A            ; Initialize counter
            SUB A               ; Sum = 0
            LXI H, 2201, H      ; Initialize pointer
    BACK  : ADD M               ; Sum = Sum + data
              INX  H            ; Increment pointer
              DCR  C            ;  Decrement counter
              JNZ BACK          ; if counter is  not equals zero  repeat
              STA 2300 H        ; Store sum
              HLT               ; Stop
```

b. Sample problem

2200 H = 04 H,
2201 H = 9 AH , 2202 H = 52 H
2203 H = 89 H ,2204 H = 3EH

Result = 9 H + 52 + 89 + 3E = 1B3 H

2300 H = B3 H, 2001 H = 01 H

Source :

```
             LDA 2200H
              MOV C, A              ; Initialize counter
              LXI  H, 2201 H        ; Initialize pointer
             SUB    A               ; Sum low = 0
            MOV B , A               ; Sum high = 0
BACK  : ADD  M                      ;  Sum = Sum + data
               JNC SKIP             ; No carry to MSB of sum
            INR  B                  ; Add carry  to MSB of sum
SKIP : INX  H                       ; increment pointer
            DCR  C                  ; Decrement counter
            JNZ BACK                ;if counter is not equals zero, repeat
            STA  2300 H             ; Store lower by the
            MOV   A, B              ;
            STA  2301 H             ; Store higher by to
            HLT
```

7.Data transfer from memory block B1 to memory block B2

Transfer ten byte of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300 H

```
Source :        MVI C, 0A H            ; Initialize counter
                LXI  H, 2200 H         ; Initialize source memory pointer
                LXI  D , 2300 H        ; Initialize destination memory pointer
            BACK  : MOV  A, M          ;Get byte form source memory
                    STAX  D            ; Store byte in the destination  memory
                    INX  H             ; Increment source memory pointer
                    INX  D             ; Increment destination memory pointer
                    DCR C              ; Decrement counter
```

```
        JNZ BACK          ; if counter is not equals zero, repeat
        HLT               ; Stop
```

8.Multiply two 8 bit numbers :

Sample problem :

2200 H= B2 H

2201 H = 03 H

Result = B2 + B2 + B2 = 216 H

2300 H = 164

2301 H = 02 H

Source:

```
        LDA   2200 H
        MOV  E, A         ; Get the first number in DE Register
        MV1 D, 00H
        LDA 2201 H ;      Get the second number in DE Register
        MOV  C, A         ; Initialize counter
        LXI H , 0000 H    ; Result = 0
BACK : DAD  D             ; Result = Result + first number
        DCR C             ; Decrement counter
        JNZ BACK          ; If check counter is not equals zero, repeat
        S HLD 2300 H      ; Store result
        HLT
```

9. Multiplication of 2 sixteen bit data

Sample problem

first data	4200 = 68	Result	4204 = 70
	4201 = 24		4205 = DE
Second data	4202 = 86		4206 = 75
	4230 = 42		4207 = 09

Source :

```
        LHLD H, 4200      ; load the first data in HL pair
        SPHL              ; More the contents to stack pointer
```

```
                LHLD 4202            ; Load the second data in HL pair
                XCHG                 ; Store the content of HL with  DE
                LXI H, 0000          ; clear the HL
                LXI B, 0000          ; clear the BC
        BACK   : DAD SP              ; Add stack pointer to HL
            JNC SKIP                 ; if carry is equal to zero jump to BACK
            INX  B                   ; increment the BC
    SKIP :      DCX D                ; Decrement the counter which  is DE
                MOV A, E             ; Move the content of E register to Acc
                ORA   D              ; Perform logical OR operation with D
                JNZ   BACK           ; Check for zero flags, if no zero   jump to  BACK.
                SHLD 4204            ; Store the content of  HL in memory location 4204
                MOV L , C            ; Move the content of C register to L
                MOV H, B             ; Move the content of B to H
                SHLD 4206            ; Store the content in memory location 4206
                HLT                  ; stop
```

10. Divide 16 bit number by 8 bit number :

Sample problem :

2200 H = 60 H 2202 H = 12

2201 H = A0 H

Result = A0 60 / 12 = 8 E 8 H , Quotient and 10 H remainder

2300 H = E 8H , 2301 H = 08 H, 2302 H = 10 H

2303 H = 00 H

Source :

```
                LHLD 2200H          ; Get the dividend
                LDA 2202 H
                MOV C, A            ; Get  the divisor
                LXI  D, 0000 H      ; quotient = 0
        BACK :  MOV A, L
                SUB C               ; subtract divisor
                MOV L , A           ; save partial result
```

```
                    JNC SKIP            ; if cy # 1 jump
                    DCH  H
        SKIP   :    INX  D              ; Increment quotient
                    MOV A,H
                    CPI 00              ; check if dividend < divisor
                    JNZ BACK            ; if no repeat .
                    MOV A, L
                    CMP C
                    JNC  BACK
                    SHLD 2302 H         ; Store the remainder
                    XCHG
                    SHLD 2300 H         ; store quotient
                    HLT
```

11. Division of 8 bit data

```
Sample problem          4200 H = 98 – Dividend
                        4201 H = 45  - Divisor
                        4202  = 02      Quotient
                        4203 = OE   – Remainder
```

```
          LDA 4201 H        ; Load the divisor
          MOV B, A          ; Move the accumulator with B  register
          LDA 4200 H        ; load the dividend
          MV1  C, 00        ; clear  the C  register to count  for quotient
BACK :    CMP   B           ; Compare the  B content with the Acc content
          JC  SKIP          ; if divisor is not less than dividend, jump
          SUB   B           ; subtract   B  register from Acc
          INR   C           ; increment the Quotient
          JMP BACK          ; jump to BACK
SKIP  :   STA 4203          ; Store the remainder at 4203
          MOV  A, C
          STA   4202 H      ; Store the Quotient at 4202 H
          HLT .
```

12. Find the largest of given number

Sample problem :

2200 H = 04, 2201 H = 34 H , 2202H = A9H,
2203 H = 78 H , 2204 H = 56 H

Result = 2300 H = A9H

Source :

```
        LDA 2200 H
        MOV  C, A           ; Initialize counter
        XRA  A              ; Maximum = minimum possible value  = 0
        LXI H, 2201 H       ; Initialize pointer
BACK  : CMP   M             ; is number > maximum
          JNC SKIP
          MOV  A, M         ; yes , replace maximum
SKIP  : INX  H
        DCR  C
        JNZ BACK
        STA  2300 H         ; Store maximum number
        HLT                 ; stop
```

13. Find the smallest of given number

Sample problem :

4200 H = 03, 4201 = 05
4202 H = 61, 4203= 09

Result 4300H = 01

```
LXI H, 4200 H       ; load the address of the first element in array
MOV  C, M           ; Initialize counter
INR H               ; Increment pointer
MOV A, M            ; Get first data in Acc
DCR  C              ; Decrement counter
```

```
BACK  INX  H            ; increment the pointer
      CMP  M            ; if number  > ACC (minimum)
      JC SKIP           ; if carry is equal to one go to SKIP
      MOV A,M           ; yes replace minimum
      DCR C             ; Decrement count
      JNZ BACK          ; if there is no zero go to BACK
      STA 4300H         ; Store the smallest data in memory at 43000
      HLT
```

14. Arrange numbers in the ascending order

```
Sample problem    2200H = 08
                  2201 H = 04
                  2202 H  = 06
Result  :         2200H  = 04
                  2201 H = 06
                  2202 H  = 08
```

Source :

```
        MVI B, 09           : Initialize counter 1
START  : LXI H, 2200 H      ; Initialize memory pointer
         MVI C, 09 H        ; Initialize counter 2
BACK  : MOV A, M            ; Get the number
        INX  H              ; increment memory pointer
        CMP M               ; is number > Acc
        JC SKIP             ; if less, don’t interchange
        JZ SKIP             ; if equal , don’t interchange
        MOV D, M
        MOV M, A
        DCX  H
        MOV  M, D           ; Interchange 2 number
        INX  H              ; increment memory pointer
SKIP   :DCR  C              ; Decrement counter 2
        JNZ BACK            ; it not zero, repeat
```

```
            DCR  B              ; Decrement counter 1
            JNZ START           ; if not zero . repeat
            HLT                 ;  Stop
```

15. Sorting an array of data in descending order

Sample problem

4200 H = 03   No of data

4201 H = 02, 4202 H=– 08 4203 H= 04

Result = 4201 H = 08, 4202H = 04 4203 H = 02

Source :

```
        LDA 4200 H
        MOV B, A      ; Initialize the counter
        DCR B         ; Decrement counter
        LXI H, 4200   ; Load the data address in to the HL
        MOV C, M      ; Move the content of memory with C
        DCR  C        ; Decrement  C register
        INX  H        ; increment pointer
BACK  : MOV A, M  ; Get the first number
        INX  H         ; Increment pointer
        CMP  M         ; it number > Acc
        JNC SKIP      ; if Carry flag is reset , jump
        MOV D, M
        MOV  M, A
        DCX  H
        MOV  M, D     ; interchange 2 number
        INX  H        ;
Skip:   DCR  C        ; Decrement the C
        JNZ BACK      ; if no zero , Jump
        HLT           ; stop
```

16. Find the number of negative, zero and positive number :

A list of 10 numbers is stored in memory starting at 4000 H. Find number of negative , zero and positive numbers from this list and store these results in memory location, 5000 H, 5001 etc .

Source :

```
            LXI  H,  400 H          ; Initialize memory pointer
            MVI C, 00H              ; Initialize no counter
            MVI  B, 00H             ; Initialize negative number counter
            MVI  E, 00H             ; Initialize  zero  number counter
BEGIN   :MOV A, M                   ; Get the number
            CPI  00 H               ; if number = 0
            JZ ZERO                 ; Go to ZERO
            ANI 80 H                ; If MSB of no = 1,if number is negative go to NEGNUM
            JNZ  NEGNUM             :
            IN R  D                 ; otherwise increment positive no counter
            JMP  LAST
ZERO  : INR  E                      ; increment Zero no counter
             JMP LAST
NEGNUM : INR  B                     ; increment negative no counter
LAST :         INX  H               ; increment memory pointer
               INR  C               ; increment number counter
                MOV A, C            ;
               CPI  0AH             ; If number counter = 10 then  Store
            JNZ BEGIN               ;  otherwise check next  no
            LXI H, 5000H
            MOV M, B                ; Store negative number
            INX  H
            MOV M, E                ; Store  no of zero
```

```
          INX  H
          MOV M, D              ; Store positive number
          HLT
```

17.Calculate the sum of series of even number

Sample problem :

2200 H = 4,   2201 H = 20

2202 H = 15, 2203 H = 13

2204 H = 22

Result = 20 + 22 = 42 H

2210 H = 42 H

Source :

```
          LDA  2200 H
          MOV  C, A             ; Initialize counter
          MVI  B, 00 H          ; Sum = 0
          LXI H, 2201 H         ; Initialize  pointer
BACK  : MOV A, M                ; Get the first number
          ANI 01 H              ; Mask Bit 1 , to Bit 7
          JNZ SKIP              ; Don’ t add if no is ODD
          MOV A, B              ; Get the sum
          ADD M                 ; Sum = sum + data
          MOV B, A              ; Store the result in B register
SKIP  : INX H                   ; Increment pointer
          DCR  C                ; Decrement counter
          JNZ BACK              ; if counter  0 repeat
          STA 2210 H            ; Store sum
          HLT                   ; Stop
```

**18.	Calculate the sum of series of odd number**

Sample program :

2200 H = 4

2201 H = 9 H   2202 H = 52

2203 H = 8H 2204 H = 3F

Result = 89 + 3F = C8

2300 H = C8 , 2301 H = 01

Source :

```
          LDA 2200 H
          MOV  C,A              ; Initialize counter
          LXI H, 2201 H         ; Initialize pointer
          MVI  E, 00            ; Sum low = 0
          MOV  D, E             ; Sum high  = 0
BACK  : MOV A, M                ; Get the first number
          ANI  01 H             ; mask bit to Bit 7
          J Z SKIP              ; Don’t add if number is even
          MOV A, E              ; Get  the lower byte of sum
          ADD  M                ; Sum  = Sum  + data
          MOV  E, A             ; Store the result in E
          JNC SKIP              ;
          INR   D               ; Add carry to MSB of sum
SKIP : INX  H                   ; Increment pointer
          DCR  C                ; Decrement counter
          JNZ  BACK             ; Check if count not equals  0 repeat
          MOV A, E
          STA 2300 H            ; Store lower byte
          MOV A, D
          STA 2301 H            ; store higher byte
          HLT                   ; Stop
```