# CHECK 1, 2, 4, 5, 6 at least.

**8085 Assembly Language Programs & Explanations**

**1. Statement**

**: Store the data byte 32H into memory location 4000H.**

Program 1: ,

MVI A, 32H : Store 32H in the accumulator

STA 4000H : Copy accumulator contents at address 4000H HLT : Terminate program execution

Program 2: LXI H : Load HL with 4000H MVI M : Store 32H in memory location pointed by HL register pair (4000H)

HLT : Terminate program execution

**2. Statement:**

**Exchange the contents of memory locations 2000H and 4000H**

Program 1:

LDA 2000H : Get the contents of memory location 2000H into accumulator

MOV B, A : Save the contents into B register

LDA 4000H : Get the contents of memory location 4000Hinto accumulator

STA 2000H : Store the contents of accumulator at address 2000H

MOV A, B : Get the saved contents back into A register

STA 4000H : Store the contents of accumulator at address 4000H

Program 2:

LXI H 2000H : Initialize HL register pair as a pointer to memory location 2000H.

LXI D 4000H : Initialize DE register pair as a pointer to memory location 4000H.

MOV B, M : Get the contents of memory location 2000H into B register.

LDAX D : Get the contents of memory location 4000H into Aregister.

MOV M, A : Store the contents of A register into memory location 2000H.

MOV A, B : Copy the contents of B register into accumulator.

STAX D : Store the contents of A register into memory location 4000H.

HLT : Terminate program execution.

**4.Statement:**

**Subtract the contents of memory location 4001H from the memorylocation 2000H and place the result in memory location 4002H.**

Program - 4: Subtract two 8-bit numbers

Sample problem: (4000H) = 51H

(4001H) = 19H

Result = 51H - 19H = 38H

Source program:

LXI H, 4000H : HL points 4000H

MOV A, M : Get first operand

INX H : HL points 4001H

SUB M : Subtract second operand

INX H : HL points 4002H

MOV M, A : Store result at 4002H.

HLT : Terminate program execution

**5.Statement:**

**Add the 16-bit number in memory locations 4000H and 4001H tothe 16-bit number in memory locations 4002H and 4003H. The most significanteight bits of the two numbers to be added are in memory locations 4001H and4003H. Store the result in memory locations 4004H and 4005H with the mostsignificant byte in memory location 4005H.**

Program - 5.a: Add two 16-bit numbers - Source Program 1

Sample problem:

(4000H) = 15H (4001H) = 1CH

(4002H) = B7H

(4003H) = 5AH

Result = 1C15 + 5AB7H = 76CCH

(4004H) = CCH

(4005H) = 76H

Source Program 1:

LHLD 4000H : Get first I6-bit number in HL

XCHG : Save first I6-bit number in DE

LHLD 4002H : Get second I6-bit number in HL

MOV A, E : Get lower byte of the first number

ADD L : Add lower byte of the second number

MOV L, A : Store result in L register

MOV A, D : Get higher byte of the first number

ADC H : Add higher byte of the second number with CARRY

MOV H, A : Store result in H register

SHLD 4004H : Store I6-bit result in memory locations 4004H and 4005H.

HLT : Terminate program execution

**6.Statement:**

**Add the contents of memory locations 40001H and 4001H and placethe result in the memory locations 4002Hand 4003H.**

Sample problem: (4000H) = 7FH

(400lH) = 89H

Result = 7FH + 89H = lO8H (4002H) = 08H

(4003H) = 0lH

Source program:

LXI H, 4000H :HL Points 4000H

MOV A, M :Get first operand

INX H :HL Points 4001H

ADD M :Add second operand

INX H :HL Points 4002H

MOV M, A :Store the lower byte of result at 4002H

MVIA, 00 :Initialize higher byte result with 00H

ADC A :Add carry in the high byte result

INX H :HL Points 4003H

MOV M, A :Store the higher byte of result at 4003H

HLT :Terminate program execution

**7.Statement:**

**Subtract the 16-bit number in memory locations 4002H and 4003Hfrom the 16-bit number in memory locations 4000H and 4001H. The mostsignificant eight bits of the two numbers are in memory locations 4001H and 4003H.Store the result in memory locations 4004H and 4005H with the most significantbyte in memory location 4005H.**

Sample problem

(4000H) = 19H

(400IH) = 6AH (4004H) = I5H (4003H) = 5CH

Result = 6A19H - 5C15H = OE04H (4004H) = 04H

(4005H) = OEH

Source program:

LHLD 4000H : Get first 16-bit number in HL

XCHG : Save first 16-bit number in DE

LHLD 4002H : Get second 16-bit number in HL

MOV A, E : Get lower byte of the first number

SUB L : Subtract lower byte of the second number

MOV L, A : Store the result in L register

MOV A, D : Get higher byte of the first number

SBB H : Subtract higher byte of second number with borrow

MOV H, A : Store l6-bit result in memory locations 4004H and 4005H.

SHLD 4004H : Store l6-bit result in memory locations 4004H and 4005H.

HLT : Terminate program execution

**8.Statement:**

**Find the l's complement of the number stored at memory location4400H and store the complemented number at memory location 4300H.**

Sample problem: (4400H) = 55H

Result = (4300B) = AAB Source program: LDA 4400B : Get the number

CMA : Complement number

STA 4300H : Store the result

HLT : Terminate program execution

**9.Statement:**

**Find the 2's complement of the number stored at memory location4200H and store the complemented number at memory location 4300H.**

Sample problem:

(4200H) = 55H

Result = (4300H) = AAH + 1 = ABH

Source program:

LDA 4200H : Get the number

CMA : Complement the number

ADI, 01 H : Add one in the number

STA 4300H : Store the result

HLT : Terminate program execution

**10.Statement:**

**Pack the two unpacked BCD numbers stored in memory locations4200H and 4201H and store result in memory location 4300H. Assume the leastsignificant digit is stored at 4200H.**

Sample problem:

(4200H) = 04 (4201H) = 09

Result = (4300H) = 94

Source program

LDA 4201H : Get the Most significant BCD digit

RLC RLC

RLC RLC : Adjust the position of the second digit (09 is changed to 90)

ANI FOH : Make least significant BCD digit zero

MOV C, A : store the partial result

LDA 4200H : Get the lower BCD digit

ADD C : Add lower BCD digit

STA 4300H : Store the result

HLT : Terminate program execution

**11.Statement:**

**Two digit BCD number is stored in memory location 4200H.Unpack the BCD number and store the two digits in memory locations 4300H and4301H such that memory location 4300H will have lower BCD digit.**

Sample problem

(4200H) = 58

Result = (4300H) = 08 and

(4301H) = 05

Source program

LDA 4200H : Get the packed BCD number

ANI FOH : Mask lower nibble

RRC RRC

RRC RRC : Adjust higher BCD digit as a lower digit

STA 4301H : Store the partial result

LDA 4200H : .Get the original BCD number

ANI OFH : Mask higher nibble

STA 4201H : Store the result

HLT : Terminate program execution

**12.Statement:**

**Read the program given below and state the contents of allregisters after the execution of each instruction in sequence.**

Main program:

4000H LXI SP, 27FFH

4003H LXI H, 2000H

4006H LXI B, 1020H

4009H CALL SUB

400CH HLT

Subroutine program:

4100H SUB: PUSH B 4101H PUSH H 4102H LXI B, 4080H

4105H LXI H, 4090H

4108H SHLD 2200H

4109H DAD B

410CH POP H

410DH POP B

410EH RET

**13.Statement:**

**Write a program to shift an eight bit data four bits right. Assumethat data is in register C.**

Source program:

MOV A, C

RAR

RAR

RAR

RAR

MOV C, A

HLT

**14.Statement**

**: Program to shift a 16-bit data 1 bit left. Assume data is in the HLregister pair**

Source program:

DAD H : Adds HL data with HL data

**15.Statement:**

**Write a set of instructions to alter the contents of flag register in8085.**

PUSH PSW : Save flags on stack

POP H : Retrieve flags in 'L'

MOV A, L : Flags in accumulator

CMA : Complement accumulator

MOV L, A : Accumulator in 'L'

PUSH H : Save on stack

POP PSW : Back to flag register

HLT :Terminate program execution

**16.Statement:**

**Calculate the sum of series of numbers. The length of the series isin memory location 4200H and the series begins from memory location 4201H.**

**a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memorylocation 4300H.**

**b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300Hand 4301H**

a. Sample problem 4200H = 04H

4201H = 10H

4202H = 45H

4203H = 33H

4204H = 22H

Result = 10 +41 + 30 + 12 = H

4300H = H

Source program:

LDA 4200H MOV C, A : Initialize counter

SUB A : sum = 0

LXI H, 420lH : Initialize pointer

BACK: ADD M : SUM = SUM + data

INX H : increment pointer

DCR C : Decrement counter

JNZ BACK : if counter 0 repeat

STA 4300H : Store sum

HLT : Terminate program execution

b. Sample problem 4200H = 04H 420lH = 9AH

4202H = 52H 4203H = 89H 4204H = 3EH

Result = 9AH + 52H + 89H + 3EH = H

4300H = B3H Lower byte

4301H = 0lH Higher byte

Source program:

LDA 4200H

MOV C, A : Initialize counter

LXI H, 4201H : Initialize pointer

SUB A :Sum low = 0

MOV B, A : Sum high = 0

BACK: ADD M : Sum = sum + data

JNC SKIP INR B : Add carry to MSB of SUM

SKIP: INX H : Increment pointer

DCR C : Decrement counter

JNZ BACK : Check if counter 0 repeat

STA 4300H : Store lower byte

MOV A, B STA 4301H : Store higher byte

HLT :Terminate program execution

**17.Statement:**

**Multiply two 8-bit numbers stored in memory locations 2200H and2201H by repetitive addition and store the result in memory locations 2300H and2301H.**

Sample problem:

(2200H) = 03H (2201H) = B2H

Result = B2H + B2H + B2H = 216H

= 216H

(2300H) = 16H

(2301H) = 02H

Source program

LDA 2200H

MOV E, A

MVI D, 00 : Get the first number in DE register pair

LDA 2201H MOV C, A : Initialize counter

LX I H, 0000 H : Result = 0

BACK: DAD D : Result = result + first number

DCR C : Decrement count

JNZ BACK : If count 0 repeat

SHLD 2300H : Store result

HLT : Terminate program execution

**18.Statement**

**:Divide 16 bit number stored in memory locations 2200H and 2201Hby the 8 bit number stored at memory location 2202H. Store the quotient in memorylocations 2300H and 2301H and remainder in memory locations 2302H and 2303H.**

Sample problem (2200H) = 60H

(2201H) = A0H

(2202H) = l2H

Result = A060H/12H = 8E8H Quotient and 10H remainder

(2300H) = E8H

(2301H) = 08H (2302H= 10H (2303H) 00H

Source program

LHLD 2200H : Get the dividend

LDA 2202H : Get the divisor

MOV C, A

LXI D, 0000H : Quotient = 0

BACK: MOV A, L

SUB C : Subtract divisor

MOV L, A : Save partial result

JNC SKIP : if CY 1 jump

DCR H : Subtract borrow of previous subtraction

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 2302H : Store the remainder

XCHG

SHLD 2300H : Store the quotient

HLT : Terminate program execution

**19.Statement:**

**Find the number of negative elements (most significant bit 1) in ablock of data. The length of the block is in memory location 2200H and the blockitself begins in memory location 2201H. Store the number of negative elements inmemory location 2300H**

Sample problem

(2200H) = 04H

(2201H) = 56H

(2202H) = A9H

(2203H) = 73H

(2204H) = 82H

Result = 02 since 2202H and 2204H contain numbers with a MSB of 1.

Source program

LDA 2200H

MOV C, A : Initialize count

MVI B, 00 : Negative number = 0

LXI H, 2201H : Initialize pointer

BACK: MOV A, M : Get the number

ANI 80H : Check for MSB

JZ SKIP : If MSB = 1

INR B : Increment negative number count

SKIP: INX H : Increment pointer

DCR C : Decrement count

JNZ BACK : If count 0 repeat

MOV A, B STA 2300H : Store the result

HLT : Terminate program execution

**20.Statement:**

**Find the largest number in a block of data. The length of the blockis in memory location 2200H and the block itself starts from memory location2201H.Store the maximum number in memory location 2300H. Assume that the numbersin the block are all 8 bit unsigned binary numbers.**

Sample problem

(2200H) = 04

(2201H) = 34H

(2202H) = A9H

(2203H) = 78H

(2204H) =56H

Result = (2202H) = A9H

Source program

LDA 2200H

MOV C, A : Initialize counter

XRA A : Maximum = Minimum possible value = 0

LXI H, 2201H : Initialize pointer

BACK: CMP M : Is number> maximum

JNC SKIP : Yes, replace maximum

MOV A, M

SKIP: INX H

DCR C

JNZ BACK

STA 2300H : Store maximum number

HLT : Terminate program execution

**21.Statement:**

**Write a program to count number of l's in the contents of Dregister and store the count in the B register.**

Source program: MVI B, 00H

MVI C, 08H

MOV A, D

BACK: RAR JNC SKIP

INR B

SKIP: DCR C

JNZ BACK

HLT

**22.Statement:**

**Write a program to sort given 10 numbers from memory location2200H in the ascending order.**

Source program:

MVI B, 09 : Initialize counter START : LXI H, 2200H: Initialize memory pointer

MVI C, 09H : Initialize counter 2

BACK: MOV A, M : Get the number

INX H : Increment memory pointer

CMP M : Compare number with next number

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

INX H : Interchange two numbers SKIP:DCR C : Decrement counter 2

JNZ BACK : If not zero, repeat

DCR B : Decrement counter 1

JNZ START HLT : Terminate program execution

**23.Statement:**

**Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you canignore carries and store the sum at memory location 2**

Sample problem:

2200H= 4H

2201H= 20H

2202H= l5H

2203H= l3H

2204H= 22H

Result 22l0H= 20 + 22 = 42H

= 42H

Source program:

LDA 2200H

MOV C, A : Initialize counter

MVI B, 00H : sum = 0

LXI H, 2201H : Initialize pointer

BACK: MOV A, M : Get the number

ANI 0lH : Mask Bit l to Bit7

JNZ SKIP : Don't add if number is ODD

MOV A, B : Get the sum

ADD M : SUM = SUM + data

MOV B, A : Store result in B register

SKIP: INX H : increment pointer

DCR C : Decrement counter

JNZ BACK : if counter 0 repeat

STA 2210H : store sum

HLT : Terminate program execution

**24.Statement:**

**Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum atmemory locations 2300H and 2301H.**

Sample problem:

2200H = 4H

2201H= 9AH

2202H= 52H

2203H= 89H

2204H= 3FH

Result = 89H + 3FH = C8H

2300H= H Lower byte

2301H = H Higher byte

Source program

LDA 2200H

MOV C, A : Initialize counter

LXI H, 2201H : Initialize pointer

MVI E, 00 : Sum low = 0

MOV D, E : Sum high = 0

BACK: MOV A, M : Get the number

ANI 0lH : Mask Bit 1 to Bit7

JZ 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 result in E register

JNC SKIP INR D : Add carry to MSB of SUM

SKIP: INX H : Increment pointer

DCR C : Decrement

**25.Statement:**

**Find the square of the given numbers from memory location 6100Hand store the result from memory location 7000H**

Source Program:

LXI H, 6200H : Initialize lookup table pointer

LXI D, 6100H : Initialize source memory pointer

LXI B, 7000H : Initialize destination memory pointer

BACK: LDAX D : Get the number

MOV L, A : A point to the square

MOV A, M : Get the square

STAX B : Store the result at destination memory location

INX D : Increment source memory pointer

INX B : Increment destination memory pointer

MOV A, C CPI 05H : Check for last number

JNZ BACK : If not repeat

HLT : Terminate program execution

**26.Statement:**

**Search the given byte in the list of 50 numbers stored in theconsecutive memory locations and store the address of memory location in thememory locations 2200H and 2201H. Assume byte is in the C register and startingaddress of the list is 2000H. If byte is not found store 00 at 2200H and 2201H.**

Source program:

LX I H, 2000H : Initialize memory pointer 52H

MVI B, 52H : Initialize counter

BACK: MOV A, M : Get the number

CMP C : Compare with the given byte

JZ LAST : Go last if match occurs

INX H : Increment memory pointer

DCR B : Decrement counter

JNZ B : I f not zero, repeat LXI H, 0000H SHLD 2200H JMP END : Store 00 at 2200H and 2201H

LAST: SHLD 2200H : Store memory address

END: HLT : Stop

**27.Statement:**

**Two decimal numbers six digits each, are stored in BCD packageform. Each number occupies a sequence of byte in the memory. The startingaddress of first number is 6000H Write an assembly language program that addsthese two numbers and stores the sum in the same format starting from memorylocation 6200H**

Source Program:

LXI H, 6000H : Initialize pointer l to first number

LXI D, 6l00H : Initialize pointer2 to second number

LXI B, 6200H : Initialize pointer3 to result

STC CMC : Carry = 0

BACK: LDAX D : Get the digit

ADD M : Add two digits

DAA : Adjust for decimal

STAX.B : Store the result

INX H : Increment pointer 1

INX D : Increment pointer2

INX B : Increment result pointer

MOV A, LCPI 06H : Check for last digit

JNZ BACK : If not last digit repeat

HLT : Terminate program execution