1. **To perform addition of two 8 bit numbers using 8085.**

ALGORITHM:

1) Start the program by loading the first data into Accumulator.
2) Move the data to a register (B register).
3) Get the second data and load into Accumulator.
4) Add the two register contents.
5) Check for carry.
6) Store the value of sum and carry in memory location.
7) Terminate the program.

PROGRAM:

MVI C, 00	Initialize C register to 00
LDA 4150	Load the value to Accumulator.
MOV B, A	Move the content of Accumulator to B register.
LDA 4151	Load the value to Accumulator.
ADD B	Add the value of register B to A
JNC LOOP	Jump on no carry.
INR C	Increment value of register C
LOOP: STA 4152	Store the value of Accumulator (SUM).
MOV A, C	Move content of register C to Acc.
STA 4153	Store the value of Accumulator (CARRY)
HLT	Halt the program.

OR

LXI H,4150H	4150(05)=HL
MOV A,M	A=05
INX H	4151(04)=HL
ADD M	A=A+M
STA 4152	

OBSERVATION:

Input: 80 (4150)
80 (4251)
Output: 00 (4152)

01 (4153)

2. To perform the subtraction of two 8 bit numbers using 8085.

ALGORITHM:

1. Start the program by loading the first data into Accumulator.
2. Move the data to a register (B register).
3. Get the second data and load into Accumulator.
4. Subtract the two register contents.
5. Check for carry.
6. If carry is present take 2's complement of Accumulator.
7. Store the value of borrow in memory location.
8. Store the difference value (present in Accumulator) to a memory
9. location and terminate the program.

PROGRAM:

MVI C, 00	Initialize C to 00
LDA 4150	Load the value to Acc.
MOV B, A	Move the content of Acc to B register.
LDA 4151	Load the value to Acc.
SUB B (02-06) 04	
JNC LOOP	Jump on no carry.
CMA	Complement Accumulator contents.
INR A	Increment value in Accumulator.
INR C	Increment value in register C
LOOP: STA 4152	Store the value of A-reg to memory address.
MOV A, C	Move contents of register C to Accumulator.
STA 4153	Store the value of Accumulator memory address.
HLT	Terminate the program.

OBSERVATION:

Input: 06 (4150)
02 (4251)
Output: 04 (4152)
01 (4153)

3.To perform the multiplication of two 8 bit numbers using 8085.

ALGORITHM:

PROGRAM:

MVI D, 00	Initialize register D to 00
MVI A, 00	Initialize Accumulator content to 00
LXI H, 4150	
MOV B, M	Get the first number in B - reg
INX H	
MOV C, M	Get the second number in C- reg.
LOOP: ADD B	Add content of A - reg to register B.
JNC NEXT	Jump on no carry to NEXT.
INR D	Increment content of register D
NEXT: DCR C	Decrement content of register C.
JNZ LOOP	Jump on no zero to address
STA 4152	Store the result in Memory
MOV A, D	
STA 4153	Store the MSB of result in Memory
HLT	Terminate the program.

OBSERVATION:

Input: FF (4150)
FF (4151)
Output: 01 (4152)
FE (4153)

4. **To perform the division of two 8 bit numbers using 8085.**

ALGORITHM:

1) Start the program by loading HL register pair with address of memory location.
2) Move the data to a register(B register).
3) Get the second data and load into Accumulator.
4) Compare the two numbers to check for carry.
5) Subtract the two numbers.
6) Increment the value of carry .
7) Check whether repeated subtraction is over and store the value of product and carry in memory location.
8) Terminate the program.

PROGRAM:

LXI H, 4150	
MOV B, M	Get the dividend in B – reg.

```
MVI C, 00          Clear C – reg for qoutient
INX H
MOV A, M           Get the divisor in A – reg.
NEXT: CMP B        Compare A - reg with register B. A<B CY=1
JC LOOP            Jump on carry to LOOP
SUB B                    Subtract A – reg from B- reg.
INR C                    Increment content of register C.
JMP NEXT           Jump to NEXT
LOOP: STA 4152     Store the remainder in Memory
MOV A, C
STA 4153           Store the quotient in memory
HLT                Terminate the program.
```

OBSERVATION:

Input: FF (4150)
FF (4251)
Output: 01 (4152) ---- Remainder
FE (4153) ---- Quotient

5. To find the largest number in an array of data using 8085 instruction set.

ALGORITHM:

1) Load the address of the first element of the array in HL pair

2) Move the count to B – reg.

3) Increment the pointer

4) Get the first data in A – reg.

5) Decrement the count.

6) Increment the pointer

7) Compare the content of memory addressed by HL pair with that of A - reg.

8) If Carry = 0, go to step 10 or if Carry = 1 go to step 9

9) Move the content of memory addressed by HL to A – reg.

10) Decrement the count

11) Check for Zero of the count. If ZF = 0, go to step 6, or if ZF = 1 go to next step.

12) Store the largest data in memory.

13) Terminate the program.

PROGRAM:

```
LXI H,4200      Set pointer for array
MOV B,M         Load the Count
INX H
MOV A,M         Set 1st element as largest data
DCR B           Decrement the count
LOOP: INX H
CMP M           If A- reg > M go to AHEAD
JNC AHEAD
MOV A,M         Set the new value as largest
AHEAD: DCR B
JNZ LOOP        Repeat comparisons till count = 0
STA 4300        Store the largest value at 4300
HLT
```

OBSERVATION:

Input: 05 (4200) ----- Array Size
0A (4201)
F1 (4202)
1F (4203)
26 (4204)
FE (4205)
Output: FE (4300)

6.To find the smallest number in an array of data using 8085 instruction set.

ALGORITHM:

1) Load the address of the first element of the array in HL pair

2) Move the count to B – reg.

3) Increment the pointer

4) Get the first data in A – reg.

5) Decrement the count.

6) Increment the pointer

7) Compare the content of memory addressed by HL pair with that of A - reg.

8) If carry = 1, go to step 10 or if Carry = 0 go to step 9

9) Move the content of memory addressed by HL to A – reg.

10) Decrement the count

11) Check for Zero of the count. If ZF = 0, go to step 6, or if ZF = 1 go to next step.

12) Store the smallest data in memory.

13) Terminate the program.

PROGRAM:

```
LXI H,4200        Set pointer for array
MOV B,M           Load the Count
INX H
MOV A,M           Set 1st element as largest data
DCR B                 Decrement the count
LOOP: INX H
CMP M                 If A- reg < M go to AHEAD
JC AHEAD
MOV A,M           Set the new value as smallest
AHEAD: DCR B
JNZ LOOP          Repeat comparisons till count = 0
STA 4300          Store the largest value at 4300
```

HLT

OBSERVATION:

Input: 05 (4200) ----- Array Size
0A (4201)
F1 (4202)
1F (4203)
26 (4204)
FE (4205)
Output: 0A (4300)

7.To write a program to arrange an array of data in ascending order
ALGORITHM:
1. Initialize HL pair as memory pointer
2. Get the count at 4200 into C – register
3. Copy it in D – register (for bubble sort (N-1) times required)
4. Get the first value in A – register
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of A –register and Memory
7. Decrement D –register content by 1
8. Repeat steps 5 and 7 till the value in D- register become zero
9. Decrement C –register content by 1
10. Repeat steps 3 to 9 till the value in C – register becomes zero

PROGRAM:

```
LXI H,4200
MOV C,M
DCR C
REPEAT: MOV D,C
LXI H,4201
LOOP: MOV A,M
INX H
CMP M
JC SKIP
MOV B,M
MOV M,A
DCX H
MOV M,B
INX H
SKIP: DCR D
JNZ LOOP
```

DCR C
JNZ REPEAT
HLT

OBSERVATION:
Input: 4200 05 (Array Size)
4201 05
4202 04
4203 03
4204 02
4205 01
Output: 4200 05(Array Size)
4201 01
4202 02
4203 03
4204 04
4205 05

8.To write a program to arrange an array of data in descending order
ALGORITHM:
1. Initialize HL pair as memory pointer
2. Get the count at 4200 into C – register
3. Copy it in D – register (for bubble sort (N-1) times required)
4. Get the first value in A – register
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of A –register and Memory
7. Decrement D –register content by 1
8. Repeat steps 5 and 7 till the value in D- register become zero
9. Decrement C –register content by 1
10. Repeat steps 3 to 9 till the value in C – register becomes zero

PROGRAM:
LXI H,4200
MOV C,M
DCR C
REPEAT: MOV D,C
LXI H,4201
LOOP: MOV A,M
INX H
CMP M 0 1 <02 A<M
JNC SKIP

```
MOV B,M     B=02
MOV M,A
DCX H
MOV M,B
INX H
SKIP: DCR D
JNZ LOOP
DCR C
JNZ REPEAT
HLT
```

OBSERVATION:
Input: 4200 05 (Array Size)
4201 01
4202 02
4203 03
4204 04
4205 05
Output: 4200 05(Array Size)
4201 05
4202 04
4203 03
4204 02
4205 01

9. Ten data bytes are stored at location starting from C100 H. Add them and store the 16 bit result at C700 H and C701 H.

```
MVI A, 00H

MOV B, A

LXI H, C100H

MOV A, M

INX H

MVI C, 04H

LOOP: ADD M

JNC NEXT
```

INR B

NEXT: INX H

DCR C

JNZ LOOP

STA C700H

MOV A, C

STA C701H

HLT

Problem – Write an assembly language program to add hexadecimal numbers stored in continuous memory or in an array.

2000	LDA 2050	A <- [2050]
2003	MOV B, A	B <- A
2004	LXI H, 2051	H <- 20 and L <- 51
2007	MVI A, 00	A <- 00
2009	MVI C, 00	C <- 00
200B	ADD M	A <- A+M
200C	INR L 2052/INX H	M <- M+1
200D	JNC 2011	
2010	INR C	C <- C+1
2011	DCR B	B <- B-1

2012	JNZ 200B	
2015	STA 3050	3050 <- A
2018	MOV A, C	A <- C
2019	STA 3051	3051 <- A
201C	HLT	Terminates the program

10.Write a program to determine number of +ve, -ve and zeros from ten signed data bytes stored at C100 H onwards. Store the answer at location C200 H onwards.

```
LXI H, C100H        (MEMORY PONTER Initialization)
MVI C, 0A H         (SIZE OF ARRAY)
MVI B, 00H          (NO. OF ZERO)
MVI D, 00H          POSITIVE
MVI E, 00H          NEGATIVE
LOOP: MOV A, M
CPI 00H
JNZ PON
INR B
JMP OVER
PON: RAL
JC NEG
INR D
JMP OVER
NEG: INR E
OVER: INX H  C1001
DCR C
JNZ LOOP
LXI H, C200H
MOV M, B
INX H
```

```
MOV M, D
INX H
MOV M, E
RST 1
```

11.Write a program to add two BCD numbers stored at C100 H and C101 H. store the answer at C200 H. and C201 H.

```
MVI C, 00H
LXI H, C100H
MOV A, M
INX H
ADD M  A=92
DAA
JNC AHEAD
INR C
AHEAD: STA C200H
MOV A, C
STA C201H
RST 1
```

12.To add two 16-bit numbers stored at consecutive memory locations

ALGORITHM:

1. Initialize memory pointer to data location.

2. Get the first number from memory and store in Register pair.

3. Get the second number in memory and add it to the Register pair.

4. Store the sum & carry in separate memory locations.

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT
8000		START	LHLD	8050H	Load the augend in DE pair through HL pair.
8001					
8002					
8003			XCHG		
8004			LHLD	8052H	Load the addend in HL pair.
8005					
8006					
8007			MVI	A, 00H	Initialize reg. A for carry
8008					
8009			DAD	D	Add the contents of HL Pair with that of DE pair.
800A			JNC	LOOP	If there is no carry, go to the instruction labeled LOOP.
800B					
800C					
800D			INR	A	Otherwise increment reg. A
800E		LOOP	SHLD	8054H	Store the content of HL Pair in 8054H(LSB of sum)
800F					
8010					
8011			STA	8056H	Store the carry in 8056H through Acc. (MSB of sum).
8012					
8013					
8014			HLT		Stop the program.

13.To write data transfer array programs and execute.

Statement: 5 no. of bytes is stored from the memory locations 2201h. Transfer the entire block of data bytes from 2201H to 2301H onwards.

Algorithm (Logic):1. Initialize the source memory pointer.
2. Initialize the destination memory pointer.
3. Initialize the counter with 5.
4. Move the contents of the source memory to accumulator.
5. Do whatever manipulation is specified /required.
6. Transfer the accumulator contents to destination memory location.
7. Increment source, destination memory pointer and decrement the counter.
8. If the count is not zero, jump back to step 4.
9. If the count is zero, stop.

Sample:
N=5 bytes, source location: 2201 to 2205, destination location: 2301 to 2305.
Data Transfer:

Memory Address	Opcode/data/address	label	Mnemonics	Comments
7000,01,02	21,01,22		LXI H,2201H	Initialize HL pair with source (src) memory.
7003,04,05	01,01,23		LXI B,2301H	Initialize BC pair with destination (destn) memory.
7006,07	16,05		MVI D,05H	Initialize D with count=05
7008	7E	LOOP	MOV A,M	Transfer src data to accumulator (acc).
7009	02		STAX B	Transfer acc contents to destn.
700A	23		INX H	Increment HLpair by 1
700B	03		INX B	Increment BC pair by1
700C	15		DCRD	Decrement D by1
700D,0E,0F	C2,08,70		JNZ LOOP	Jump to loop if Zero flag is not set.
7010	CF		RST1	Stop

Data: 5 no. of bytes stored from 2201 onwards.

Result: Same 5 no. of bytes stored in 2301 onwards

15. Write a program to generate the Fibonacci SERIES.

1. Initialize register H with 30 and register L with 50, so that indirect memory M points to memory location 3050.
2. Initialize register B with 00, register C with 08 and register D with 01.
3. Move the content of B in M.
4. Increment M by 1 so that M points to next memory location.
5. Move the content of D in M.
6. Move the content of B in accumulator A.
7. Add the content of D in A.
8. Move the content of D in B.
9. Move the content of A in D.
10. Increment M by 1 so that M points to next memory location.
11. Move the content of A in M.
12. Decrements C by 1.
13. Jump to memory location 200C if ZF = 0 otherwise Halt the program.

Program –

Program

Address	HEX Codes	Labels	Mnemonics	Comments
8000	21, 50, 80	START	LXI H ,8050H	Pointer to the OUT-BUFFER
8003	AF		XRA A	Clear accumulator and reg. B
8004	47		MOV B, A	
8005	77		MOV M, A	Copying content to the target location
8006	3C		INR A	Increment A
8007	23		INX H	Go to the next dest. address.
8008	77		MOV M, A	Moving the content
80 09	0E, 08		MVI C, 08H	Initialize counter
800B	80	LOOP	ADD B	Getting the next term
800C	46		MOV B, M	Initializing term e.g. F1 = F2
800D	23		INX H	Go to the next dest. address.
800E	77		MOV M, A	Writing to the OUT-BUFFER

Address	HEX Codes	Labels	Mnemonics	Comments
800F	0D		DCR C	Decrement count until 0 is reached F3= F1 + F2 (A) = (A) + (B) This is done with instruction ADDB.
8010	C2, 0B, 80		JNZ LOOP	
8013	76		HLT	Terminate the program

17.To find the square of the number from 0 to 9 using a Table of Square.
ALGORITHM:
1. Initialize HL pair to point Look up table
2. Get the data .
3. Check whether the given input is less than 9.
4. If yes go to next step else halt the program
5. Add the desired address with the accumulator content
6. Store the result

Program

Address	HEXCodes	Labels	Mnemonics	Comments
F000	21,00, 90		LXI H, 9000H	Point to the lookup table address
F003	3A,00, 80		LDA 8000H	Get the data
F006	FE,0F		CPI 0FH	Checkinput > 15D

Address	HEXCodes	Labels	Mnemonics	Comments
F008	DA,13, F0		JC AFTER	The check the number greater than 0A or not
F00B	3E, FF		MVI A, FFH	Load FFH into A
F00D	32,50, 80		STA 8050H	Store FFH for numbers > 15D
F010	C3,1B, F0		JMP DONE	End the program
F013	4F	AFTER	MOV C, A	Add the desired Address
F014	06,00		MVI B,00H	Clear register B
F016	09		DAD B	ADD BC with HL pair
F017	7E		MOV A, M	Take the result from Look-up table
F018	32,50, 80		STA 8050H	Store the result
F01B	76	DONE	HLT	Terminate the program

LOOKUP TABLE:
4125 01
4126 04
4127 09
4128 16
4129 25
4130 36
4131 49
4132 64
4133 81

OBSERVATION:
Input: 4150: 05
Output: 4151 25 (Square)
Input : 4150: 11
Output: 4151: FF (Error Indication)

18. write an 8085 ALP to count the no. of byte that are greater than 25_{10} and lesser than 65_{10} from an array of twenty bytes stored on M.L. 2000h onwards , store such no. on M.L. 2100H. onwards.

```
LXI H, 2000H

LXI D,2100H

MVI C, 14H(20MDECIMAL)

MVI B, 00H

X12:MOV A,M

    CPI  19H(25 DECIMAL)

    JC  X11

   CPI 41H(65 DECIMAL) A

JNC X11

INR B

STAX D

INX D

X11: INX H

DCR C

JNZ X12

HLT
```

19.write-a-8085-program-to-count-the-number-of-even-numbers.

```
1. LXI H, 1200H : Initialize memory pointer 1, points to list of
   50 numbers
2. LXI D, 1300H : Initialize memory pointer 2, points to list of
   numbers which will contain even numbers
3. MVI C, 32H : Initialize counter
4. BACK: MOV A, M : Get the number
5. ANI 0lH : Check for even number
6. JNZ SKIP : If ODD, don't store JZ A=0
7. MOV A, M : Get the number
8. STAX D : Store the number in result list
9. INX D : Increment pointer 2
10. SKIP: INX H : Increment pointer l
11. DCR C : Decrement counter
12. JNZ BACK : If not zero, repeat
13. HLT
```

A=A ANI 01=00

). **An array of ten numbers is stored from memory location 2000H onwards. Write an 8085 assembly language program to separate out and store the EVEN and ODD numbers on new arrays from 2100H and 2200H, respectively.**

```
LXI H 2000H

LXI D 2100H

LXI B 2200H

MVI A 0AH

COUNTER: STA 3000H

MOV A M

ANI 01H

JNZ CARRY

MOV A M

STAX B

INX B

JMP JUMP

CARRY: MOV A M ; This block will store Odd numbers.

STAX D

INX D

JUMP: LDA 3000H

DCR A

INX H

JNZ COUNTER

HLT
```

```
LXI B,1400H

LXI D,1300H

MVI C, 0AH

BACK: MOV A,M
```

```
ANI 01H
JNZ SKIP
MOV A, M
STAX D
INX D
JMP NOW
SKIP:MOV A,M
STAX B
INX B
NOW:INX H
DCR C
JNZ BACK
HLT
```

20. AN array of twenty data bytes is stored on M.L. 4100H onwards. Write an ALP to remove the duplicate entries from array & sore the compresses array on a new array storing from M.L. 4200H ONWARD

```
MVI B, 0AH
   MVI C, 01H
   LXI H, 4100H
   SHLD 3000H
   LDA 4100H
   STA 4110H
L1: LHLD 3000H
MOV A, M
```

```
INX H
DCR B
JZ OVER
SHLD 3000H
LXI H, 4110H
MOV D, C
L2: CMP M
JZ L1
INX H
DCR D
JNZ L2
MOV M, A
INR C
JMP L1
OVER: HLT
```

An array of ten data bytes is stored on memory locations 2100H onwards. Write an 8085 assembly language program to find the bytes having complemented nibbles (e.g. 2DH, 3CH, 78H etc.) and store them on a new array starting from memory locations 2200H onwards.

2D=0010 1101

78-0111 1000

```
LXI H, 2100h
LXI D,2200H
L2:MOV B,M
```

```
MOV A,B
ANI 0FH
MOV C,A
MOV A,B
ANI F0H
RRC
RRC
RRC
RRC
XRA C
CPI 0FH
JNZ L1
MOV A,B
STAX D
INX D
L1:INX H
MOV A,L
CPI 0AH
JNZ L2
HLT
```