Assignment 2

1. Write an assembly program to take two numbers as input and output their sum using registers.

MVI A, 25H

MVI B, 1AH

ADD B

STA 3000H

HLT

2. Write an assembly program to take two numbers as input and output their sum using memory.

START: LXI H, 2000H

MOV A, M

LXI D, 2001H

MOV B, M

ADD B

LXI H, 3000H

STA 3001H

HLT

3. Write an assembly program to take two numbers as input and output their difference using registers.

MVI A, 35H

MVI B, 15H

SUB B

STA 3000H

HLT

4. Write an assembly program to take two numbers as input and output their difference using memory.

LXI H, 2000H

MOV A, M

LXI H, 2001H

SUB M

STA 3000H

HLT

5. Write an assembly program to take two numbers as input and output their product using registers.

MVI B, 25H

MVI A, 00H

MVI D, 08H

MVI C, 05H

LOOP2: ADD B

JNC LOOP

INR D

LOOP: DCR C

JNZ LOOP2

STA 2005H

HLT

6. Write an assembly program to take two numbers as input and output their product using memory.

LXI H, 2000H

MOV B, M

MVI A, 00H

MVI D, 00H

INX H

MOV C, M

LOOP2: ADD B

JNC LOOP

INR D

LOOP: DCR C

JNZ LOOP2

STA 2005H

HLT

7. Write an assembly program to take two numbers as input and output their quotient using registers.

MVI B, 06H

MVI A, 00H

MVI D, 00H

INX H

MVI C, 02H

MOV A, B

LOOP: SUB C

INR D

CMP C

JNC LOOP

STA 2005H

MOV A, D

STA 2006H

HLT

8. Write an assembly program to take two numbers as input and output their quotient using memory.

LXI H, 2000H

MOV B, M

MVI A, 00H

MVI D, 00H

INX H

MOV C, M

MOV A, B

LOOP: SUB C

INR D

CMP C

JNC LOOP

STA 2005H

MOV A, D

STA 2006H

HLT

Assignment 3

1. Write an assembly program to display numbers from 1 to 5 using a loop.

MVI A, 01H

MVI B, 01H

MVI C, 05H

LXI D, 0000H

LOOP: STAX D

INX D

ADD B

DCR C

JNZ LOOP

HLT

2. Write an assembly program to calculate the factorial of a given number.

LXI H, 2000H

MOV B, M

MVI D, 01H

FACTORIAL: CALL MULTIPLY

DCR B

JNZ FACTORIAL

INX H

MOV M, D

HLT

MULTIPLY: MOV E, B

MVI A, 00H

MULTIPLYLOOP: ADD D

DCR E

JNZ MULTIPLYLOOP

MOV D, A

RET

3. Develop an assembly program to check if a given number is odd or even using branching statements.

LDA 2050H

RAR

JC LOOP

MVI A, 02H

STA 3050H

HLT

LOOP: MVI A, 01H

STA 3050H

HLT

4. Develop an assembly program to determine if a given number is positive or negative using branching.

LXI H, 5FFFH

MOV C, M

MVI B, 00H

MVI D, 00H

MVI E, 00H

LOOP3: INX H

MOV A, M

ADI 00H

JZ LOOP1

JP LOOP2

INR E

JMP LOOP4

LOOP1: INR D

JMP LOOP4

LOOP2: INR B

LOOP4: DCR C

JNZ LOOP3

MOV A, B

STA 7001H

MOV A, E

STA 7002H

HLT

5. Write an assembly language program to find the smaller of two numbers stored in memory locations 2501H and 2502H. Store the result in 2503H memory location.

LXI H, 2501H

MOV A, M

INX H

MOV B, M

CMP B

JC LOOP

MOV A, B

LOOP: STA 2503H

HLT

6. Write an assembly program to determine whether a given year is a leap year or not using conditional branches.

LDA 2050H

RAR

JC LOOP

RAR

JC LOOP

MVI A, 02H

STA 2051H

HLT

LOOP: MVI A, 01H

STA 2051H

HLT

7. Write an assembly program to use unconditional jump instructions for program flow.

MVI A, 05H

JMP LOOP

DCR A

LOOP: STA 0000H

HLT

8. Write an assembly program to use conditional jump instructions for program flow.

MVI A, 05H

LXI D, 0000H

LOOP: STAX D

INX D

DCR A

JNZ LOOP

HLT

Assignment 4

1. Write an assembly program to shift bits to the left.

LDA 3000H

RAL

STA 3001H

HLT

2. Write an assembly program to shift bits to the right.

LDA 3000H

RAR

STA 3001H

HLT

3. Write an assembly program to rotate bits to the left.

LDA 3000H

RLC

STA 3001H

HLT

4. Write an assembly program to rotate bits to the right.

LDA 3000H

RRC

STA 3001H

HLT

5. Write an assembly program to generate a series of odd numbers up to a specified limit.

LXI H, 2000H

MOV C, M

MVI A, 01H

MVI B, 02H

LXI D, 2001H

LOOP: STAX D

INX D

ADD B

CMP C

JC LOOP

HLT

6. Write an assembly program to generate a series of prime numbers within a specified range.

LXI H, 2000H

MOV A, M

MVI C, 00H

MOV D, A

MOV E, A

L2: MOV B, D

L1: CMP B

JC LABEL

SUB B

JNZ L1

LABEL: CPI 00H

JNZ SKIP

INR C

SKIP: MOV A, E

DCR D

JNZ L2

MOV A, C

STA 2001H

HLT

7. Write an assembly program to implement a function to check if a given number is a palindrome or not.

LDA 8000H

MOV H, A

MVI C, 08H

LOOP: MOV A, H

RLC

MOV H, A

MOV A, D

RAR

MOV D, A

DCR C

JNZ LOOP

MOV A, H

CMP D

JZ TRUE

MVI A, 00H

JMP EXIT

TRUE: MVI A, FFH

EXIT: STA 8050H

HLT

8. Write an assembly program to implement a function to find the greatest common divisor (GCD) of two numbers.

MVI A, 06H

MVI B, 09H

CMP B

JZ DOWN

JNC SHIFT

MOV C, A

MOV A, B

MOV B, C

SHIFT: SUB B

CMP B

JZ MOVE

JNC SHIFT

MOV C, A

MOV A, B

MOV B, C

JMP SHIFT

MOVE: MOV A, B

DOWN: STA 0000H

HLT

9. Write an assembly program to find the least common multiple (LCM) of two numbers.

LXI H, 0010H

MOV C, M

MVI B, 00H

INX H

MOV A, M

CMA

MOV E, A

MVI D, FFH

INX D

LXI H, 0000H

NEXT: DAD B

SHLD 0050H

LOOP: DAD D

JNC SKIP

MOV A, H

ORA L

JZ EXIT

JMP LOOP

SKIP: LHLD 0050H

JMP NEXT

EXIT: LHLD 0050H

HLT

10. Write an assembly program to develop a function to perform the bubble sort algorithm on an integer array.

START: LXI H, 0000H

MVI D, 00H

MOV C, M

DCR C

INX H

FLAG: MOV A, M

INX H

CMP M

JC NEXT

JZ NEXT

MOV B, M

MOV M, A

DCX H

MOV M, B

INX H

MVI D, 01H

NEXT: DCR C

JNZ FLAG

MOV A, D

CPI 01H

JZ START

HLT

11. Write an assembly program to develop a function to compute the power of a number (x^n).

LXI H, 0000H

MOV B, M

INX H

MOV C, M

MVI D, 01H

POWERLOOP: CALL MULTIPLY

DCR C

JNZ POWERLOOP

INX H

MOV M, D

HLT

MULTIPLY: MOV E, B

MVI A, 00H

MULTIPLYLOOP: ADD D

DCR E

JNZ MULTIPLYLOOP

MOV D, A

RET

12. Write an assembly program to calculate the sum of digits in a number using a function.

LDA 0000H

MOV C, A

ANI 0FH

MOV B, A

MOV A, C

ANI F0H

RRC

RRC

RRC

RRC

ADD B

STA 0010H

HLT

Assignment 5

1. Design an assembly program that functions as a simple calculator, performing basic arithmetic operations (addition, subtraction, multiplication and division) based on user input.

LDA 0000H

CPI 01H

JZ ADDITION

CPI 02H

JZ SUBTRACTION

CPI 03H

JZ MULTIPLICATION

CPI 04H

JZ DIVISION

JMP END

ADDITION: LDA 0001H

LXI H, 0002H

MOV B, M

ADD B

STA 0003H

JMP END

SUBTRACTION: LDA 0001H

LXI H, 0002H

MOV B, M

SUB B

STA 0003H

JMP END

MULTIPLICATION: MVI A, 00H

STA 8003H

LXI H, 0001H

MOV B, M

LXI H, 0002H

MOV C, M

MLOOP: ADD B

DCR C

JNZ MLOOP

STA 0003H

JMP END

DIVISION: MVI B, 00H

LDA 0001H

LXI H, 0002H

MOV C, M

DLOOP: INR B

SUB C

JNZ DLOOP

MOV A, B

STA 0003H

HLT

JMP END

END: HLT