BRANCHING INSTRUCTION SET

1. Write a program for the addition of two 8 bit numbers with carry.

PROGRAM STATEMENT

To write an assembly language program for the addition of two 8 bit numbers with carry.

PROGRAM ANALYSIS

- ALGORITHM:

  STEP 1: Load the data into accumulator from memory location ~~2000~~ from 2000H.

  STEP 2: Move the data into B register

  STEP 3: Load the data into accumulator from memory location 2001H.

  STEP 4: Add the registers.

  STEP 5: Check for carry.

  STEP 6: Store the value of sum and carry in memory location 2002H and 2003H respectively.

  STEP 7: Terminate the program.

ASSEMBLY LANGUAGE CODE

```
MVI C,00H    //Intialize C register to 00H
LDA 2000H    //Load the value of memory location 2000H to accumulator.
MOV B,A      //Move the content of accumulator to register B
LDA 2001H    // Load the value of memory location 2001H to accumulator.
ADD B        // Add the content of accumulator with register B.
JNC X        // Jump on no carry to X.
INR C        // Increament content of register C by 1.
X: STA 2002H // Store the value of accumulator to memory location
                2002H (Sum)
MOV A,C      // Move the content of register C to accumulator.
STA 2003H    // Store the value of accumulator to memory location 2003H(Carry)
HLT          // Terminate the program.
```

Input:

Memory Location:

2000H ⟵ 9AH

2001H ⟵ 9FH

Register:

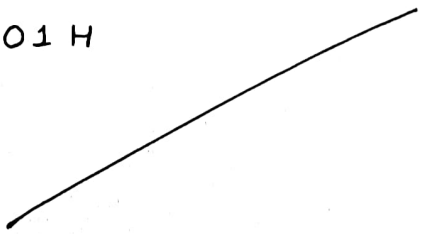
B ⟵ 9AH

A ⟵ 9FH

Output:

2003H ⟶ 39H

2004H ⟶ 01H



2. Write a program for the addition of two 16-bit numbers with carry.

PROGRAM STATEMENT

To write a assembly language program for the addition of two 16-bit numbers with carry.

PROGRAM ANALYSIS

~~ST~~ ALGORITHM:

STEP 1: Load 16 bit data (immediate) on register pair HL and DE respectively

STEP 2: Add the register pair DE and HL directly.

STEP 3: Check for carry

STEP 4: Store the sum and carry in memory location 2055H and 2060H respectively.

STEP 6: Terminate the program

ASSEMBLY LANGUAGE CODE

```
MVI C, 00H      // Initialize C with data 00H.
LXI H, 9A9FH    // Load 16 bit data to HL pair immediately.
LXI D, 9FA2H    // Load 16 bit data to DE pair immediately.
DAD D           // Double add register pair HL and DE directly.
JNC X           // Jump to X if no carry
INR C           // Increament content of register C by 1
X: SHLD 2055H   // Store the content of HL pair to mem. location 2055H.
MOV A, C        // Move content of register C to accumulator
STA 2060H       // Store the content of accumulator to ~~reg~~ 2060H
HLT             // Terminate the program
```

Input

HL ← 9A9FH

DE ← 9FA2H

Output

2055H → 41H 2056H → 30H

2060H → 01H

3. Write to add the content of 9101H and 9102H and place the sum at 9103H and carry at 9104H.

PROGRAM STATEMENT

To write an assembly language program to add content of memory location 9101H and 9102H and place the sum at 9103H and carry at 9104H.

PROGRAM ANALYSIS

ALGORITHM:

STEP 1: Initialize to store the carry bit.
STEP 2: Load data to accumulator from memory location 9101H.
STEP 3: Move the content of accumulator to register B.
STEP 4: Load data to accumulator from memory location 9102H.
STEP 5: Add the content of accumulator and register B.
STEP 6: Check for carry or not?
a) If No,
i) Store the result in a memory location 9103H and carry at 9104H.
ii) Terminate the program
b) If Yes,
i) Increase the content of register C by one.
ii) Store the addition result in memory location 9103H.
iii) Store the carry in memory location 9104H.
STEP 7: Terminate the program.

ASSEMBLY LANGUAGE CODE

```
MVI C, 00H    // Initialize C register to 00H.
LDA 9101H     // Load the content of memory location 9101H to
                 accumulator.
MOV B,A       // Move the content of accumulator to register B
LDA 9102H     // Load the content of memory location 9102H to
                 accumulator.
ADD B         // Add the content of register B with accumulator.
X: STA 9103H  //
JNC X         // Jump to X if not carry.
INR C         // Increment content of register C by 1.
X: STA 9103H  // Store the content of accumulator to memory
                 location 9103H.
MOV A, C      // Move content of register C by 01H.
STA 9104H     // Store the content of accumulator to memory location
                 9104H.
HLT           // Terminate the program
```

Input

9101H ← 7FH

9102H ← 5BH

B ← 7FH

A ← 5BH

Output

9103H → DAH

9104H → 00H

4. WAP for the subtraction of two 8-bit number with borrow.

PROGRAM STATEMENT

To write an assembly language program for the subtraction of two 8-bit number with borrow.

PROGRAM ANALYSIS

ALGORITHM:

STEP 1: Initialize register C to store borrow bit.

STEP 2: Load the data in accumulator from memory location 2000H.

STEP 3: Move the content of accumulator to register B.

STEP 4: Load the data to accumulator from memory location 2001H.

STEP 5: Subtract content of accumulator and register B.

STEP 6: If carry is present take 2's complement of accumulator.

STEP 7: Store the value of borrow in memory location 2004H.

STEP 8: Store the diffence value in memory location 2003H.

STEP 9: Terminate the program

ASSEMBLY LANGUAGE CODE

```
MVI C, 00H      // Initialize 00H to register C.
LDA 2000H       // Load the content of memory location 2000H to
                   accumulator A.
MOV B, A        // Move the content of accumulator to register B.
LDA 2001H       // Load the content of memory location 2001H to
                   accumulator A.
SUB B           // Subtract content of accumulator and register B.
JNC X           // Jump to x if no carry (x)
CMA             // Complement accumulator contents.
INR A           // Increment value in Accumulator
INR C           // Increment value in Register C.
X: STA 2002H    // Store the value of accumulator to memory address
                   2002H.
MOV A, C        // Move content of
STA 2003H       // Store the value of accumulator to memory location
                   2003H.
HLT             // Terminate the program.
```

INPUT

2000H ← 1FH

2001H ← 9AH

B ← 1FH

A ← 9AH

OUTPUT

2002H → 7BH

2003H → 00H

6. Write to subtract the content of register C and register D and store the register H and borrow at register L.

PROGRAM STATEMENT

To write an assembly language program to subtract the content of register C and register D and store the difference at H and borrow at register L.

PROGRAM ANALYSIS

ALGORITHM:

STEP 1: Initialize register B to store borrow bit.

STEP 2: Load the data to register C immediately.

STEP 3: Move the content of register C to accumulator.

STEP 4: Load the data to register D immediately.

STEP 5: Subtract content of register C and register D.

STEP 6: If carry is present take 2's complement of accumulator.

STEP 7: Store the difference value in register H.

STEP 8: Store the value of borrow in register L.

STEP 9: Terminate the program.

ASSEMBLY LANGUAGE CODE

```
MVI B, 00H   // Load 00H data on register B to store
                borrow bit.
MVI C, 35H   // Load 35H data on register C immediately.
MOV A, C     // Move the content of register C to accumulator.
MVI D, 78H   // Load 78H data on register D immediately.
SUB D        // Subtract the content of register A with
                register D.
JNC X        // Jump to X if not carry (borrow).
CMA          // Complement content of A
INR A        // Increment content of A by 01H.
INR B        // Increment content of B by 01H.
X: MOV H, A  // Move content of accumulator to register H.
MOV A, B     // Move content of register B to accumulator.
MOV L, A     // Move content of accumulator to register L.
HLT          // Terminate the program.
```

INPUT

C ← 35H

D ← 78H

Output

H → 43H

L → 01H

7. CWAP to create a continuous loop using unconditional Jump location.

PROGRAM STATEMENT

To write an assembly language program to create a continuous loop using unconditional JUMP location.

PROGRAM ANALYSIS

ALGORITHM:

STEP 1: Load Accumulator with data 10H immediately.

STEP 2: Load register B with data 01H immediately.

STEP 3: Subtract content of accumulator with register B.

STEP 4: Jump to Y if the result on accumulator is zero.

STEP 5: Jump to X.

STEP 6: Move content of accumulator to register C.

STEP 7: Terminate the program

ASSEMBLY LANGUAGE CODE

```
MVI, A, 10H   // Load 10H value to accumulator immediately.
X: MVI B, 01H // Load 01H data value to register B immediately.
SUB B         // Subtract the content of accumulator with register B.
JZ Y          // Jump to Y if content of accumulator is zero.
JMP X         // Unconditional Jump to X.
Y: MOV C, A   // Move the content of accumulator to register C.
HLT           // Terminate the program.
```

INPUT

A ← 10H

B ← 01H

OUTPUT

C → 00H

8. Write a program to test the Jump on zero condition on an arithmetic operation.

PROGRAM STATEMENT

To write a assembly language program to test the Jump on zero Condition on an arithmetic operation.

PROGRAM ANALYSIS

ALGORITHM:

STEP 1 : Load immediate value into accumulator.
STEP 2 : Load immediate value into register B
STEP 3 : Load immediate value into register C.
STEP 4 : Add contents of register B to accumulator.
STEP 5 : Compare contents of accumulator and register C.
STEP 6 : If accumulator is equal to register C, jump to label X.
STEP 7 : Load immediate value 11H into register E } No Decision Case
STEP 8 : Move content of register E to accumulator.
STEP 9 : Terminate the program
STEP 10 : Load immediate value 22H into register D } Yes Decision Case
STEP 11 : Move content of register D to accumulator.
STEP 12 : Terminate the program

ASSEMBLY LANGUAGE CODE

- For Satisfied Condition

```
MVI A, 05H   // Load immediate value 05H into accumulator.
MVI B, 02H   // Load immediate value 02H into register B.
MVI C, 07H   // Load immediate value 07H into register C.
ADD B        // Add content of accumulator with register B.
CMP C        // Compare contents of accumulator and register C.
JZ X         // If accumulator is equal to register C, jump to
               label X.
MVI E, 11H   // Load immediate value 11H into register E.
MOV A, E     // Move content of register E to accumulator.
HLT          // Terminate the program
X: MVI D, 22H // Load immediate value 22H into register D.
MOV A, D     // Move content of register D to accumulator
HLT          // Terminate the program
```

Input

A ← 05H

B ← 02H

C ← 07H

Output

A → 22H

- For Unsatisfied Condition

```
MVI A, 05H
MVI B, 02H
MVI C, 01H
ADD B
CMP C
```

```
JZ X
MVI E, 11H
MOV A, E
HLT
X: MVI D, 22H
MOV A, D
HLT
```

Input

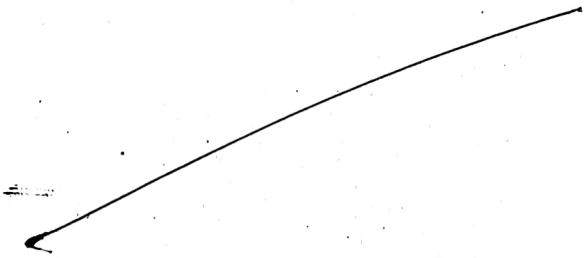
A ← 05H

B ← 02H

C ← 03H

Output

A → 11H



9. Write a program to test the Jump on Even Parity condition on logical operation.

PROGRAM STATEMENT

To write an assembly language program to test the Jump on Even Parity condition on logical operation

PROGRAM ANALYSIS

ALGORITHM:

STEP 1: Load immediate value into accumulator

STEP 2: Load immediate value into register B.

STEP 3: Load immediate value into register C.

STEP 4: AND content of register B with accumulator.

STEP 5: Compare contents of accumulator and register C.

STEP 6: If the parity is even, jump to label X.

STEP 7: Load immediate value 11H into register E.

STEP 8: Mode content of register E to accumulator.

STEP 9: Terminate the program.

STEP 10: Load immediate value 22H into register D.

STEP 11: Mae content of register D to accumulator.

STEP 12: Terminate the program

ASSEMBLY LANGUAGE CODE

- For Satisfied Condition

```
MVI A, 05H    // Load immediate value 05H into accumulator.
MVI B, 02H    // Load immediate value 02H into register B.
MVI C, 07H    // Load immediate value 07H into register C.
ANA B         // AND content of accumulator with register B.
CMP C         // Compare content of accumulator and register C.
JPE X         // If the parity is even, jump to label X.
MVI E, 11H    // Load immediate value 11H into register E.
MOV A, E      // Move content of register E to accumulator.
HLT           // Terminate the program.
X: MVI D, 22H // Load immediate value 22H into register D.
MOV A, D      // Move content of register D to accumulator.
HLT           // Terminate the program.
```

Input

A ← 05H
B ← 02H
C ← 07H

Output

A → 22H

- For Unsatisfied Condition

```
MVI A, 02H
MVI B, 02H
MVI C, 07H
ANA B
CMP C
JPE X
```

```
MVI E, 11H
MOV A, E
HLT
X: MVI D, 22H
MOV A, D
HLT
```

Input

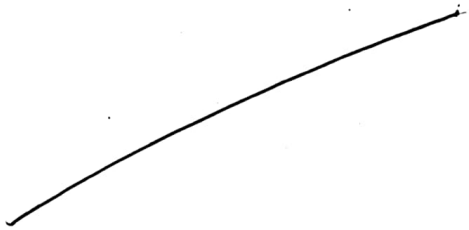
A ← 02 H

B ← 02 H

C ← 07 H

Output

A → 11H



10. Write a program to test the Jump on Negative condition on an arithmetic operation.

PROGRAM STATEMENT

To write an assembly language program to test the Jump on Negative condition on an arithmetic operation.

PROGRAM ANALYSIS

Algorithm:

Step 1: Load immediate value into accumulator.

Step 2: Load immediate value into register B.

Step 3: Load immediate value into register C.

Step 4: Subtract the content of accumulator and register B.

Step 5: Jump to label x if the content of accumulator is negative.

Step 6: Move the content of register C to accumulator.

Step 7: Terminate the program.

Step 8: Load immediate value into accumulator.

Step 9: Move the content of register D to accumulator.

Step 10: Terminate the program.

ASSEMBLY LANGUAGE CODE

- For Satisfied Condition

```
MVI A, 02H    // Load immediate value 02H into accumulator.
MVI B, 05H    // Load immediate value 05H into register B.
SUB B         // Subtract the content of register B with accumulator.
JM X          // Jump to label X if the content of
MVI C, 00H    // Load immediate value 00H into accumulator.
MOV A, C      // Move content of register C to accumulator
HLT           // Terminate the program.
X: MVI D, 01H // Load immediate value 01H into accumulator.
MOV A, D      // Move content of register D to accumulator.
HLT           // Terminate the program.
```

Input

A → 02H

B → 05H

Output

A → 01H

- For Unsatisfied Condition

```
MVI A, 05H
MVI B, 02H
SUB B.
JM X
MVI C, 00H
MOV A, C
HLT
```

X'. MVI D, 01H

MOV A, D

HLT

INPUT

A → 05H

B → 02H

OUTPUT

A → 00H