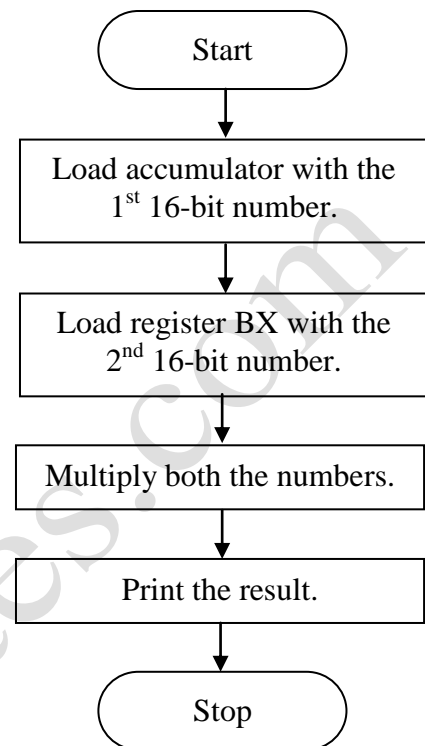


Program 16: Multiply two 16-bit signed numbers.**Program:**

Instructions	Comments
include "emu8086.inc"	
ORG 100h	
MOV AX, 0004H	Move 1 st 16-bit number to AX.
MOV BX, FFFEh	Move 2 nd 16-bit number to BX.
IMUL BX	Multiply BX with AX and the result will be in DX:AX.
CALL PRINT_NUM	Print the result.
RET	Return.
DEFINE_PRINT_NUM	Declare function.
END	

Flowchart:**Explanation:**

- This program multiplies two 16-bit signed numbers.
- The program has been developed using *emu8086* emulator available at: www.emu8086.com.
- ORG 100h is a compiler directive. It tells compiler how to handle the source code.
- It tells compiler that the executable file will be loaded at the offset of 100h (256 bytes).
- The 1st 16-bit number 0004H is a positive number and is moved to accumulator AX.
- The 2nd 16-bit number FFFEh is a negative number (-2 in decimal) and is moved to register BX.
- Then, both the numbers are multiplied.
- The multiplication of two 16-bit numbers may result into 32-bit number. So, the result is stored in the DX and AX register.
- The MSB is stored in DX and LSB is stored in AX.
- The result is printed on the screen.

Output:**Before Execution:**

AX = 0004H

BX = FFFEh (-2 in decimal)

After Execution:

AX = FFF8H (-8 in decimal)

DX = FFFFH