IA32 Floating point unit

The floating point unit has 8 register st0 to st7 that are formatted in the form of stack .Number are pushed onto the stack from memory , and popped up back to stack back to memory. Floating point unit instruction generally will pop the first two items off the stack , perform function on them and then push the answer back on to top of the stack

To increase the speed and efficiency of real-number computations, computers or FPUs typically Represent real numbers in a binary floating-point format. In this format, a real number has three Parts: a sign, a significant, and an exponent The sign is a binary value that indicates whether the number is positive (0) or negative(1)

For Floating-Point Add we use FADD for Floating-Point Subtract we use FSUB, for Floating-Point Multiply we use FMUL , for Floating-Point Divides we use FDIV.

FLD is use to push value to st register and FADD is use to pop those valus then add them and push the answer to st register at 0.

**Code segment**

**1)**

include irvine32.inc

.data

num real4 5.04

num1 real4 3.0

.code

main proc

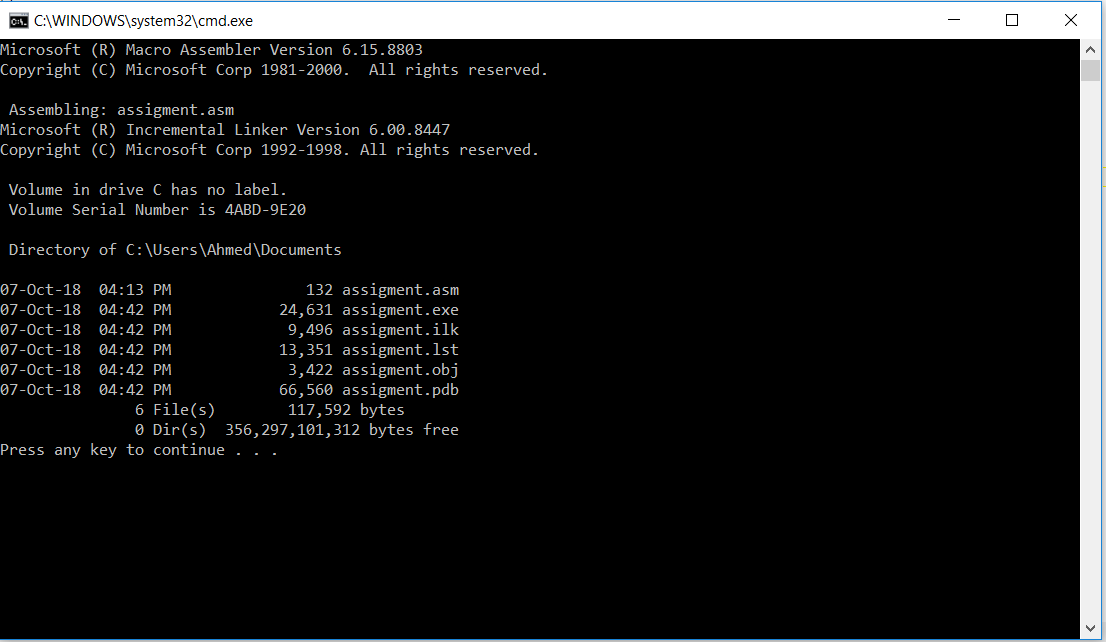
fld num

fld num1

FADD

main endp

end main



**2)**

include irvine32.inc

.data

num real4 5.04

num1 real4 3.0

.code

main proc

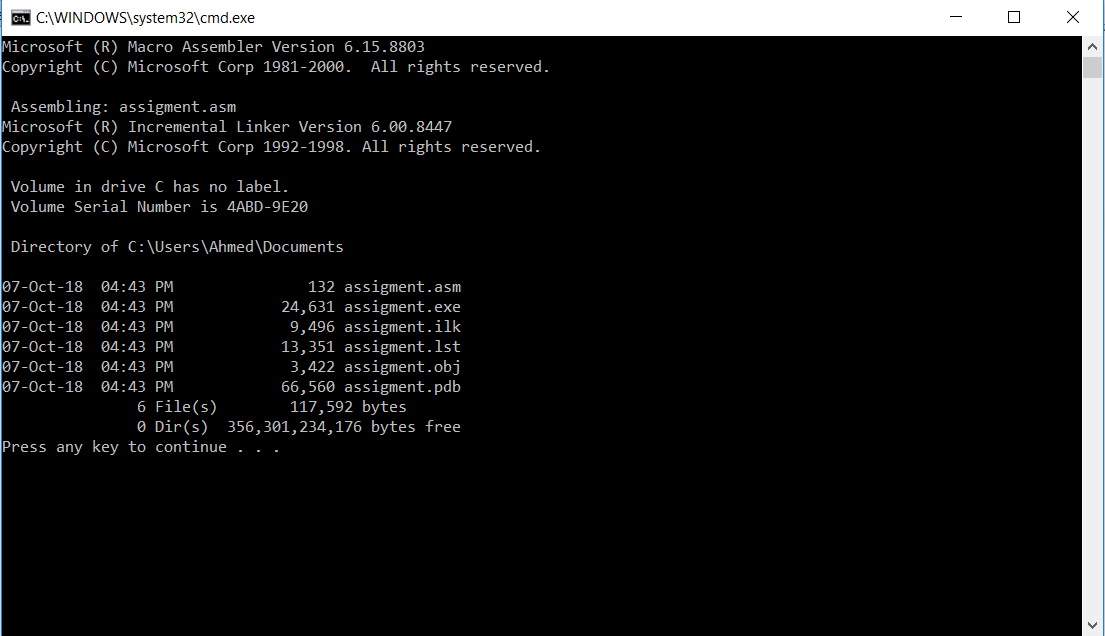
fld num

fld num1

FSUB

main endp

end main



**3)**

include irvine32.inc

.data

num real4 5.04

num1 real4 3.0

.code

main proc

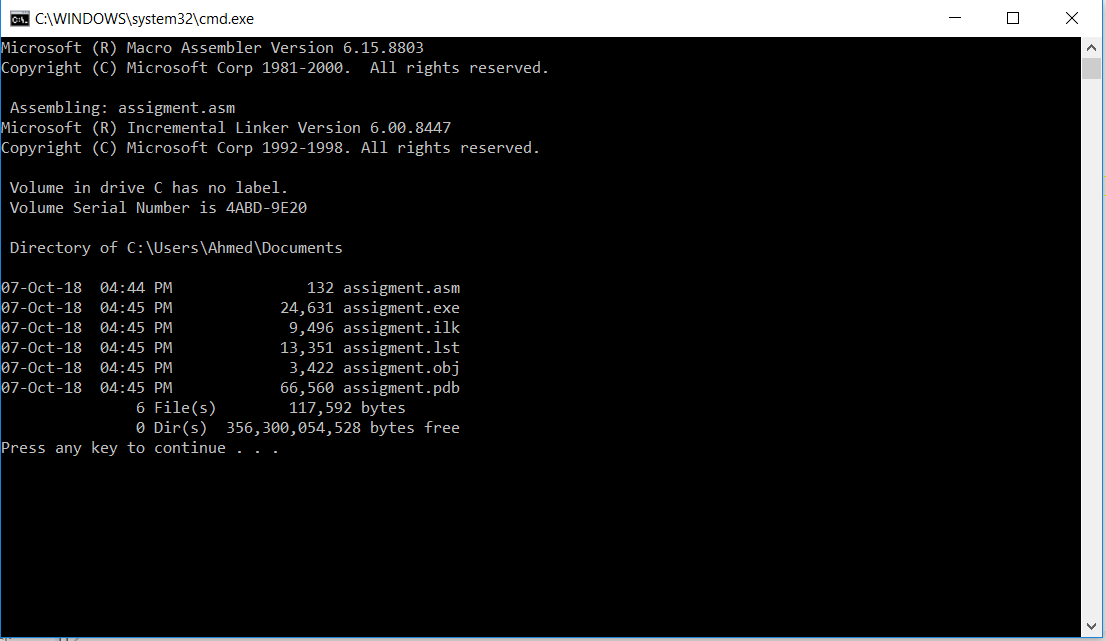
fld num

fld num1

FMUL

main endp

end main



**4)**

include irvine32.inc

.data

num real4 5.04

num1 real4 3.0

.code

main proc

fld num

fld num1

FDIV

main endp

end main

