**CSE2006-LAB\_10**

**NAME:** O G RAGAVI

**REG NO:** 20BCE1988

**EX NO:** 10

**TITLE: 8087 instructions**

**GENERAL :**

**finit** - initialise the coprocessor to start the beginning of the stack

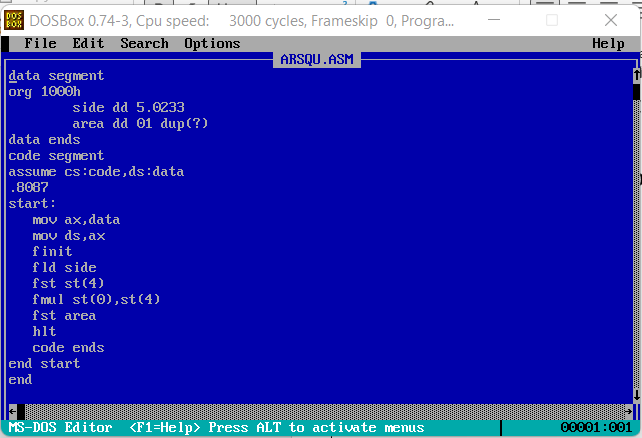
**fld X** – load value X onto the top of the stack

1.TO FIND THE AREA OF THE SQUARE WHOSE SIDE IS A FLOATING POINT NUMBER USING 8087.

**ALGORITHM:**

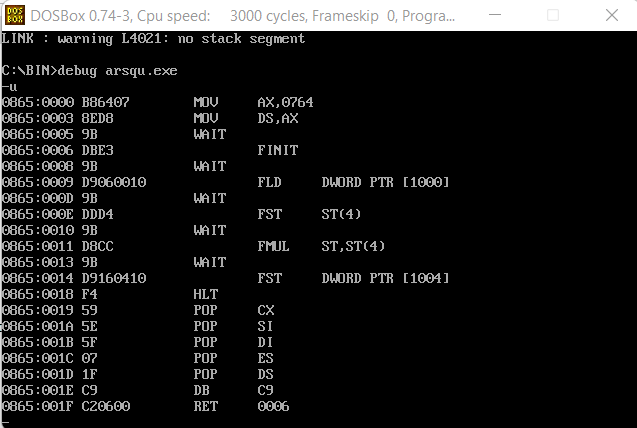
* Inside the code segment, initialise the start address of the stack (here 1000h) and the side value . Reserve I byte of storing the result value into the variable ‘squarear’.
* Inside the code segment, initialise 8087(co-processor) by .8087 instruction and move the data into the DX register.
* Initialise the 8087 and load the side value onto the top of the stack.
* Store the side value in the 4th segment of the stack( for instance) and multiply with the side value which is stored in the 0th segment of the stack( so it is side\*side).
* Store the result i.e side\*side into the variable ‘squarear’.
* Terminate the program successfully.

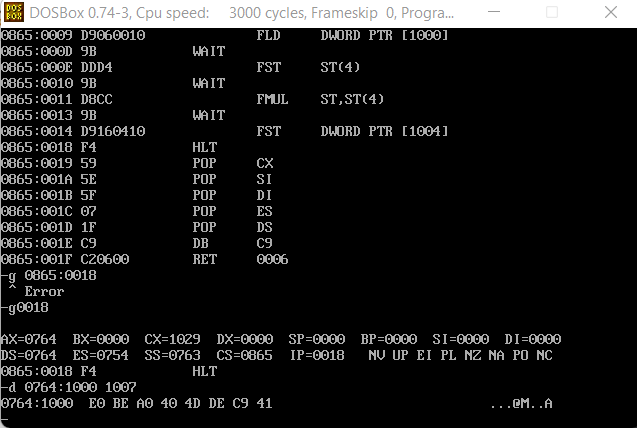
**CODE:**

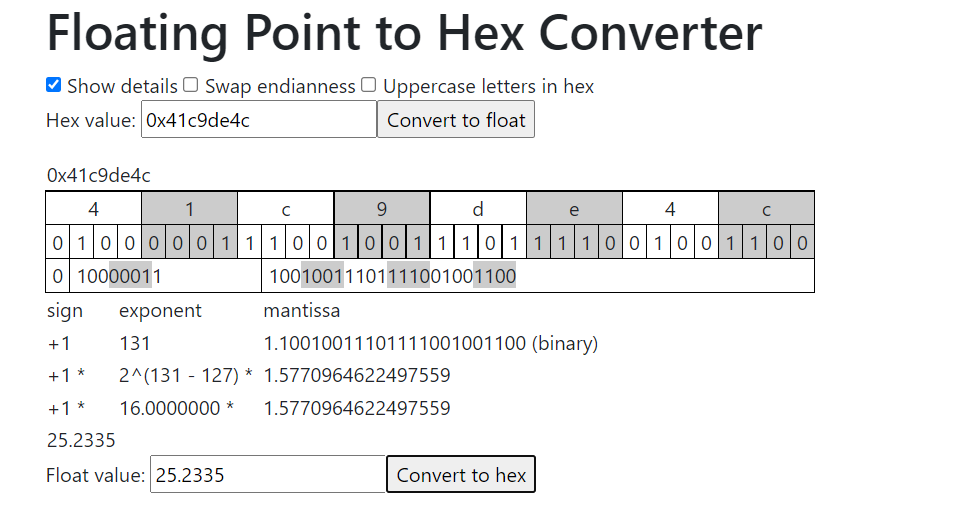
****

**INPUT:** 5.023

**OUTPUT:**

****

****

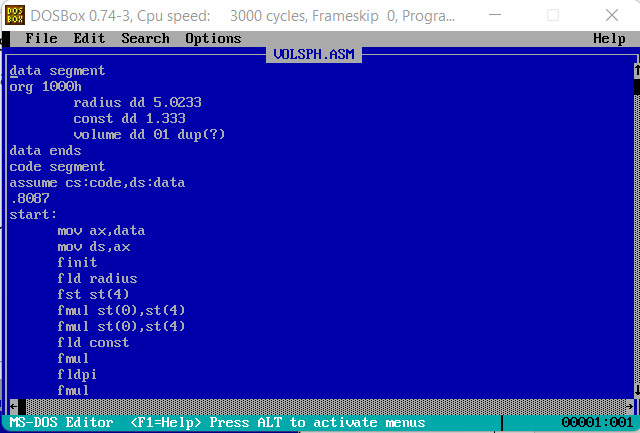
****

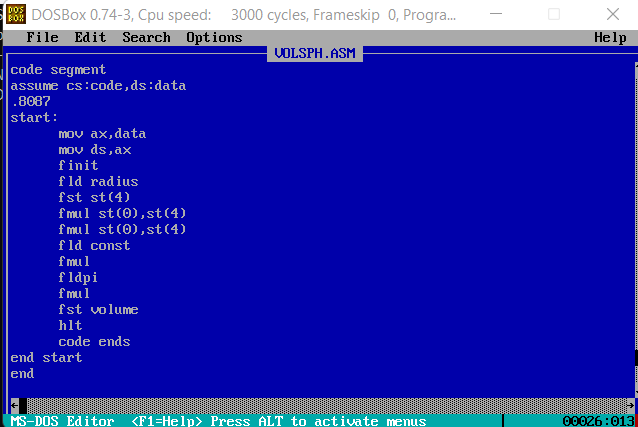
2.TO FIND THE VOLUME OF A SPHERE WHOSE SIDE IS A FLOATING POINT NUMBER USING 8087.

**ALGORITHM:**

* Inside the code segment, initialise the start address of the stack (here 1000h) the radius value and the const value(here 4/3=1.333). Reserve I byte of storing the result value into the variable ‘volume’.
* Inside the code segment, initialise 8087(co-processor) by .8087 instruction and move the data into the DX register.
* Initialise the 8087 and load the radius value onto the top of the stack.
* Store the side value in the 4th segment of the stack( for instance) and multiply with the side value which is stored in the 0th segment of the stack( so it is radius\*radius).Again multiply the value the value stored in 4th segment of the stack with the top of the stack, so that it becomes radius \* radius \* radius.
* Load the value of the constant onto the top of the stack and multiply it with the value already stored in the top(radius \* radius \* radius).Also load the value of the pi and do the same.
* Store the result i.e constant\*pi\*radius\*radius\*radius into the variable ‘volume’.
* Terminate the program successfully.

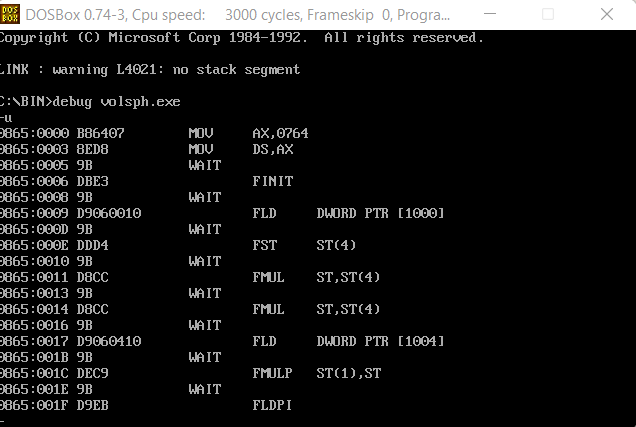
**CODE:**

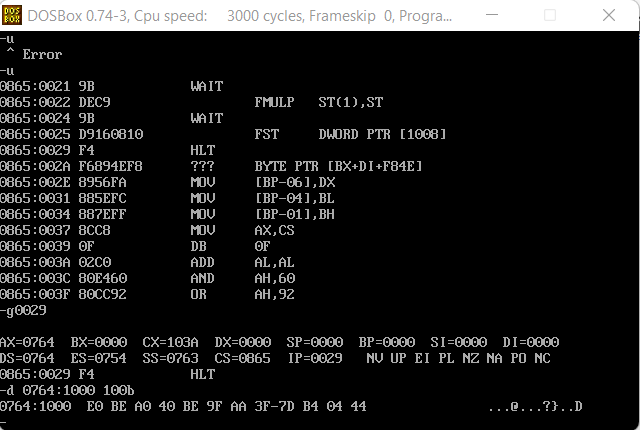
****

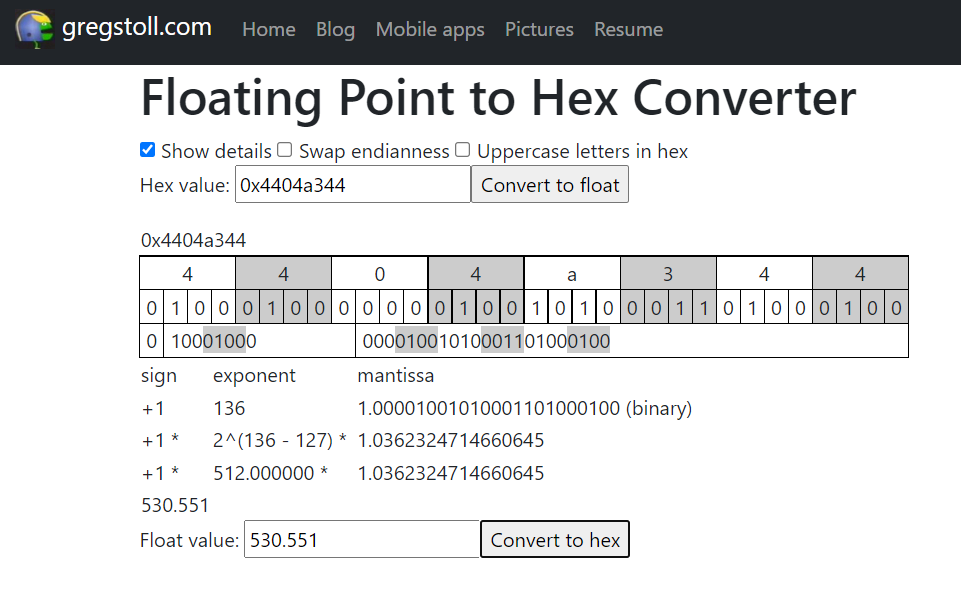
****

**INPUT: radius =5.0233**

**OUTPUT:**

****

****

****

**RESULT:**

Hence, the use of 8087 as a co-processor to handle floating point numbers is being understood using the ALP program done above.