**Lab report #9**

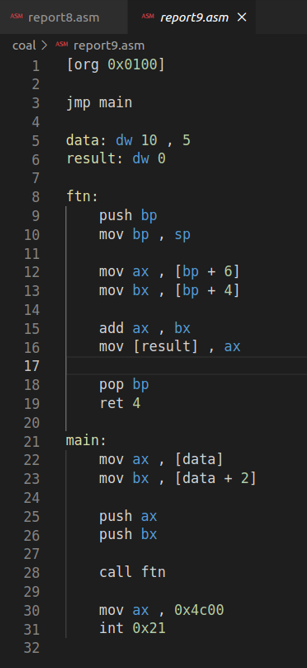
**Saad Ahmad**

**(20P-0051)**

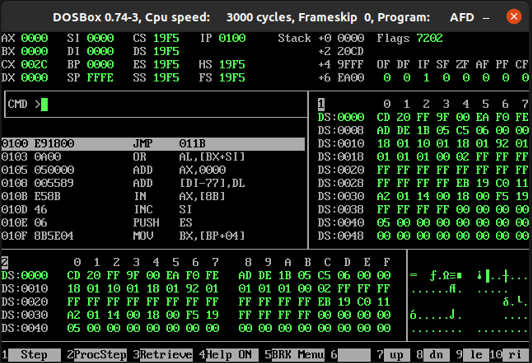
**Subroutines with parameters:**

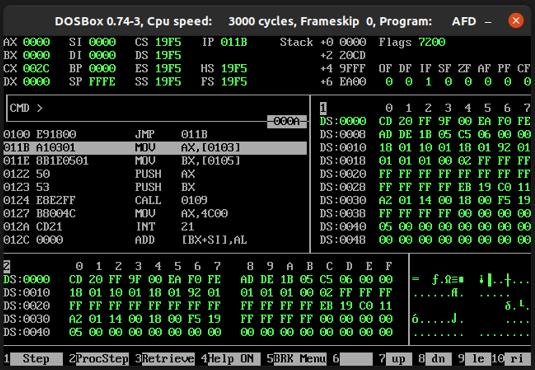
The concept of subroutine with parameter is a little bit different with the high-level language’s functions. In parameter subroutine the parameter is actually stored in stack as we have the limited registers, as in stack there is the concept of first in and last out so with the help of stack, we can store our parameters on it and can easily access it by base pointer. And when we end our subroutine we will add a number with the ret keyword which will clean the stack once our code is executed.

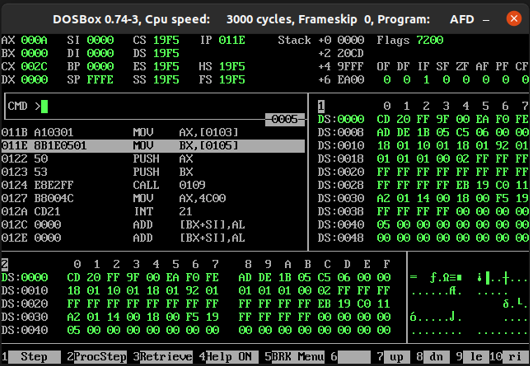
**Code:**

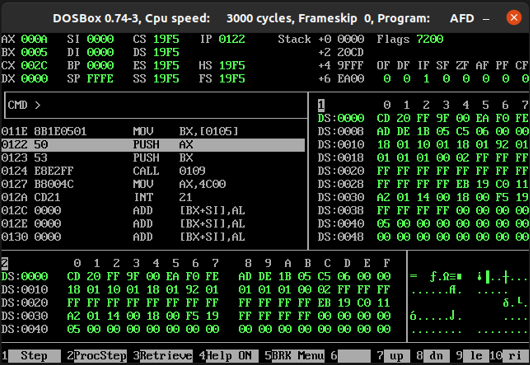


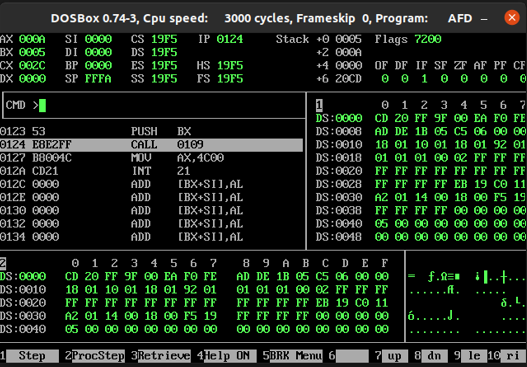
**Output:**

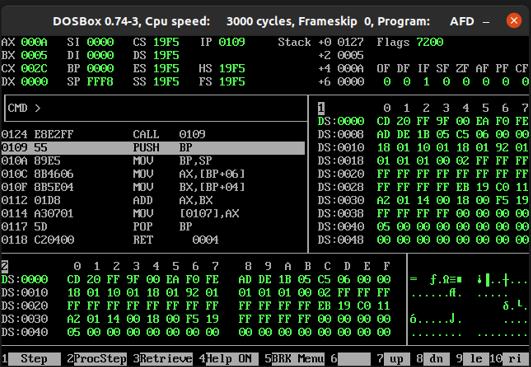


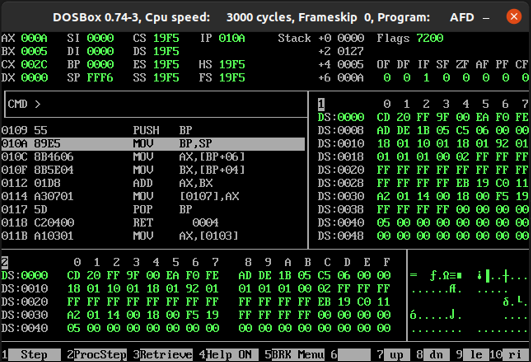


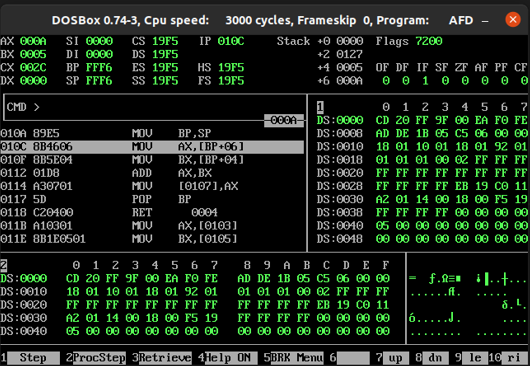


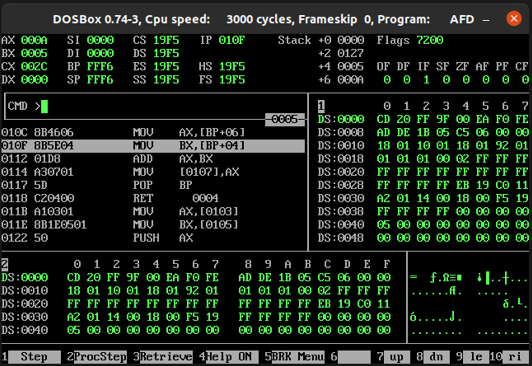


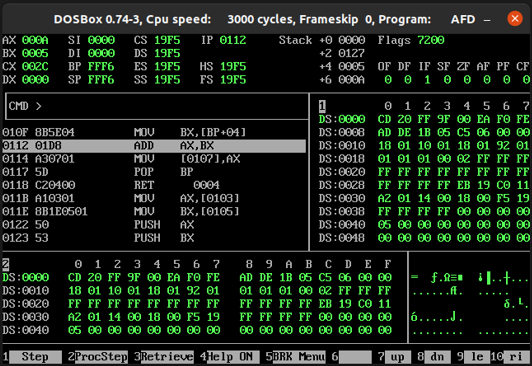


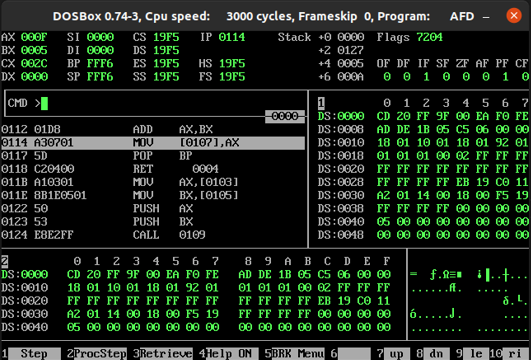


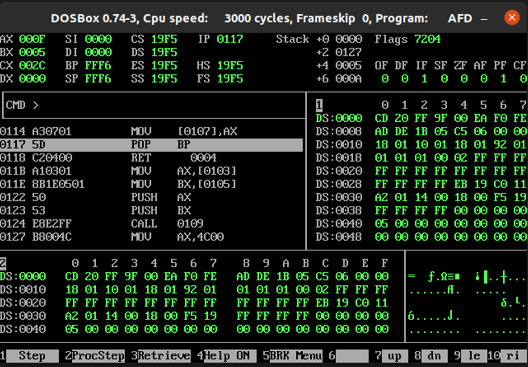


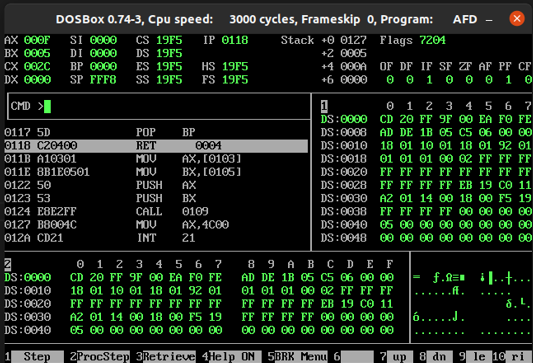


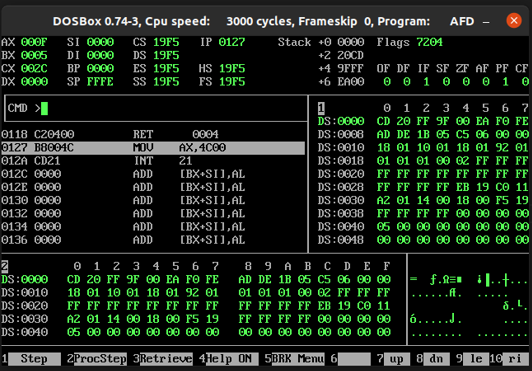


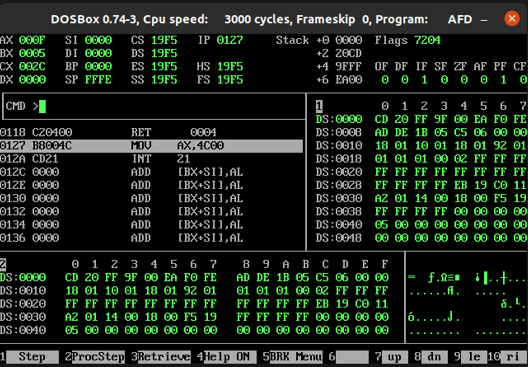


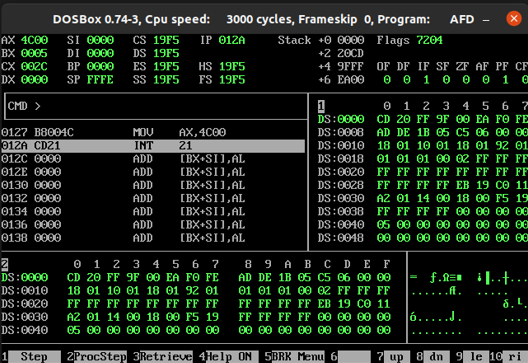


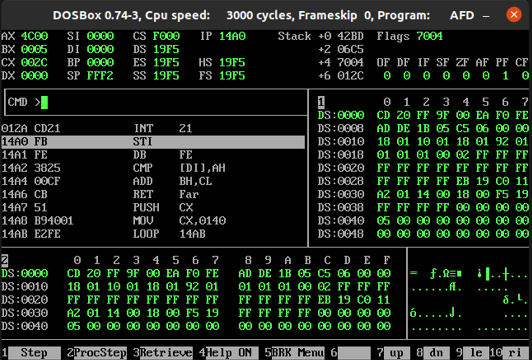








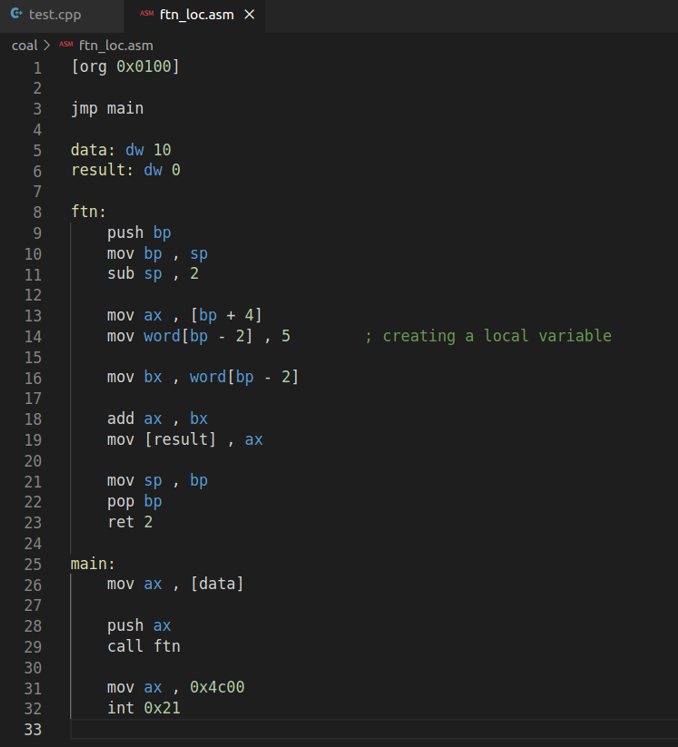




**LOCAL VARIABLES**

An important role of the stack is in the creation of local variables that are only needed while the subroutine is in execution and not afterwards. They should not take permanent space like global variables. Local variables should be created when the subroutine is called and discarded afterwards. So that the spaced used by them can be reused for the local variables of another subroutine. They only have meaning inside the subroutine and no meaning outside it.

**Code:**



**Output:**

