Text

Description automatically generated

Pruebas:

1. Avance6\_1 *pasa*
2. Avance6\_3 *pasa*
3. Avance6\_6 *pasa*
4. Avance7\_1 [ accesar index negativo] *pasa pero porque no consideré negativos al definir mi léxico…*
5. Avance7\_2 [accesar index fuera de rango] *pasa*
6. Avance3\_3 [de todo, incluido llamadas a funciones; suma de dos llamadas distintas a la misma función; dec y acceso a arreglos] *pasa*
7. factorial [factorial cíclico y recursivo] *pasa*

**Máquina virtual: Ejecución de módulos y arreglos**

*Arreglos*

Si dirección está en el rango de direcciones de pointers (3666-3999), convertirla a la dirección real y sacar el valor

*Módulos*

*Text

Description automatically generated*

*Graphical user interface, text, application, email

Description automatically generated*

**Graphical user interface, text, application

Description automatically generated**

Crear un execution\_stack donde vaya guardando los contextos creados por llamadas a funciones.

En cada iteración del while se debe checar si el stack está vacío. Si no está vacío, se cambia current\_memory

Asimismo crear un ip\_stack donde se guarden las posiciones de los instruction pointers.

ERA:

* Crear instancia de Memory con la info de l\_op
* Añadir instancia a execution\_stack

PARAMETER:

GOSUB:

* Guardar current\_memory en temp
* Cambiar current\_memory a la de execution\_stack.pop()
* Append temp a execution\_stack
* Append instruction\_pointer+1 a ip\_stack
* Instruction\_pointer = rightop virtual address-1

ENDFUNC:

* Current\_memory = execution\_stack.pop()
* Instruction\_pointer = ip\_stack.pop()

**Me falta hacer la parte de read del avance 6**

**Quizá tenga que cambiar los rangos de virtual addresses. Por ahora, hay 333 espacios para cada variable, pero los arrays pueden ocupar hasta 10,000 espacios… Entonces en mis pruebas no puedo crear arrays tan grandes, por precaución.**