Diseño de Compiladores

Documentación de Compilador:
BIXO

Juan Camilo Granados
A01023421

Alberto Treviño
A00824493

Fecha de entrega:
13 de Junio de 2023

Descripción del Proyecto..... 5
Propósito y Alcance.....5
Análisis de Requerimientos.....5
Descripción de los Casos de Prueba..... 5
Bitácora Semanal.....6
Semana 1.....6
Semana 2.....6
Semana 3.....6
Semana 4.....6
Semana 5.....6
Semana 6.....6
Semana 7.....7
Lista de Commits.....7
Reflexión..... 1
Descripción del Lenguaje.....1
Descripción del Compilador..... 1
Equipo, lenguaje y utilerías..... 1
Descripción del Analizador Léxico..... 1
Descripción del Analizador Sintactico..... 1
Descripción de Generación de Código Intermedio y Análisis Semántico..... 1
Program.....1
Function.....1
DecFuncType..... 1
DecFunc..... 1
VoidFunction.....1
Dec-Var.....1
Type.....1
MainFunction..... 1
Modules..... 1
Body..... 1
Parameters..... 1
Expressions..... 1
Factor..... 1
Statement..... 1
Assign.....1
Read..... 1
Print..... 1
Variables.....1
Call..... 1
If..... 1
While..... 1
FuncEsp..... 1

Array....1
Arrvalues....1
Matrix....1
Matvalues....1
Mean....1
Layers....1
Sequential....1
Compile....1
Fit....1
Predict....1
Getweights....1
Fibonacci....1
Factorial....1
MatrixMult....1
Descripción de la Administración de Memoria en Compilación....1
Directorio de Funciones....1
Ejemplo Directorio de funciones....1
Pilas y Cubo Semántico....1
Descripción de la Máquina Virtual....1
Objeto VM....1
Manejo de Memoria de Ejecución....1
Pruebas del funcionamiento del lenguaje....1
Prueba.bixo....1
Código....1
Cuádruplos....1
Orden de Ejecución de Cuádruplos....1
Resultados....1
Prueba2.bixo....1
Código....1
Cuádruplos....1
Orden de Ejecución de Cuádruplos....1
Resultados....1
Pruebaif1.bixo....1
Código....1
Cuádruplos....1
Orden de Ejecución de Cuádruplos....1
Resultados....1
Pruebaif2.bixo....1
Código....1
Cuádruplos....1
Orden de Ejecución de Cuádruplos....1
Resultados....1
Pruebawhile.bixo....1
Código....1

Cuádruplos..... 1
Orden de Ejecución de Cuádruplos..... 1
Resultados..... 1
Pruebarray.bixo..... 1
Código..... 1
Cuádruplos..... 1
Orden de Ejecución de Cuádruplos..... 1
Resultados..... 1
PruebaMatrix.bixo..... 1
Código..... 1
Cuádruplos..... 1
Orden de Ejecución de Cuádruplos..... 1
Resultados..... 1
Fibonacci.bixo..... 1
Código..... 1
Cuádruplos..... 1
Orden de Ejecución de Cuádruplos..... 1
Resultados..... 1
Factorial.bixo..... 1
Código..... 1
Cuádruplos..... 1
Orden de Ejecución de Cuádruplos..... 1
Resultados..... 1
Pruebafuncesp.bixo..... 1
Código..... 1
Cuádruplos..... 1
Orden de Ejecución de Cuádruplos..... 1
Resultados..... 1

Descripción del Proyecto

Propósito y Alcance

El propósito de este proyecto es crear un compilador mediante un nuevo lenguaje de programación con sus respectivas funcionalidades, tales como ciclos, creación de variables, arreglos, condicionales, etc. Mediante el analizador léxico, sintáctico y semántico se llevará a cabo su respectiva generación de código para ejecutar el proyecto.

Análisis de Requerimientos

- Statements
- Math Expressions
- Modules
- Arrays
- Special Functions
- Loops

Descripción de los Casos de Prueba

Nombre	Descripción
prueba.bixo	Checa el funcionamiento de la ejecución de llamada a funciones con parámetros múltiples con read y write
prueba2.bixo	Comprueba el orden de operaciones binarias y el correcto funcionamiento del return a partir de la ejecución de funciones de tipo non void.
prueba3.bixo	Comprueba el funcionamiento de las funcionalidades agregadas de Arrays, con asignación y print de espacios específicos de arreglos. Implementa while con asignación dinámica.
prueba4.bixo	Comprueba el funcionamiento de las funcionalidades agregadas de Matrix, con asignación y print de espacios específicos de matrices, y multiplicación matricial.
pruebaif1.bixo	Checa las operaciones aritméticas y el funcionamiento del if/else en un caso falso.
pruebaif2.bixo	Checa las operaciones aritméticas, utilizando el operador “&” y el funcionamiento del if/else en un caso

Nombre	Descripción
	verdadero.
pruebawhile.bixo	Comprueba el funcionamiento del ciclo while utilizando el operador “&” y “|” , iterando sobre un valor hasta que la condición sea falsa.
pruebarray.bixo	Prueba el funcionamiento de arrays, su despliegue y el cálculo de su mean, y la implementación del find.
pruebaMatrix.bixo	Checa el almacenamiento de matrices de dos dimensiones junto con la multiplicación , impresión, el cálculo del mean de las mismas, y la implementación del sort.
fibonacci.bixo	Realiza el cálculo de un fibonacci.
factorial.bixo	Realiza el cálculo de un factorial.
pruebafuncesp.bixo	Comprueba el funcionamiento de las funciones especiales de machine learning y análisis de datos, y gráfico de datos.

Bitácora Semanal

Semana 1

- Se implementó el lexer y el parser

Semana 2

- Se empezó a trabajar el cubo semántico.

Semana 3

- Se arreglaron detalles del lexer y parser

Semana 4

- Se implementó la tabla de variables y el directorio de procedimientos.
- Se agregaron rangos de memoria
- Función void y normal
- Correcciones parser sintaxis

Semana 5

- Se agregaron los puntos neurálgicos para el if y while
- Se agrego el codigo para el if

- corrección del generador de cuadruplos y la pila de cuádruplos
- Ejecución correcta de variables locales y globales
- Debugging
- Merge del branch

Semana 6

- Ajustes Lexer
- Merge if y while
- Validación que no se repitan funciones
- Assign funcional
- Implementación de tabla de constantes
- Cuadruplos funcionales
- Operadores funcionales con cuadruplos
- Parámetros funcionales
- Mainfunc agregado
- Return funciones tipo int y float

Semana 7

- Implementación de la máquina virtual
- Asignación llamadas de funciones normales y recursivas
- If else completado
- Temporales con contador
- Tabla de temporales con dirección de memoria
- Cuadruplos comparativos
- While
- Arrays
- Declaración de matrices
- Funciones especiales
- Fibonacci

Lista de Commits

Se obtuvieron los commits mediante el comando git log:

commit aa6cac6fa95aaac11eca78bdc4c3ac19af56acb4
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon Jun 5 01:40:32 2023 -0600

ajuste prueba matrix

commit c74aa78bd610f37ef81aa5b9f01cd4b096330ab1
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon Jun 5 01:22:25 2023 -0600

```
    ajuste pruebas

commit 127ac1e146dd3115025aaaa4d0955be0e3d6c6e1
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Mon Jun 5 01:02:42 2023 -0600

    ajuste pruebas

commit 6e075cf836eb9357669889a11255f10daa672b77
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Mon Jun 5 00:40:35 2023 -0600

    ajuste prueba2

commit 8e173696137e0497d91525a4e6458eb2823845a6
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun Jun 4 19:01:28 2023 -0600

    ajuste parser funcesp

commit b34173e8cf94efd0287cb7782933c96003ebbcaf
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun Jun 4 18:57:11 2023 -0600

    funciones gosub y todo funcionando

commit 73c6014ad8174e860d58dd5abe3dfcce5e57fcef
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun Jun 4 15:46:55 2023 -0600

    arregle el while

commit 4c6c6ef2bc22a4acce176fd2748b761e82ccaa6f
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun Jun 4 14:30:09 2023 -0600

    ajuste pruebawhile

commit 853df0d7ac052008cf94d21e11879bd420350587
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun Jun 4 14:24:41 2023 -0600

    cambios pruebawhile

commit b38081fc8332a07f7c56d41795f6a3bcd6443c6c
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sun Jun 4 12:05:01 2023 -0600
```

```
    ajustes

commit 49094816e2b73156c19de3947e161fdfd5539a97
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sun Jun 4 11:56:30 2023 -0600

    ajustes

commit a8204c074e4dcb7b17b2625a905fa4f819d5b030
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 19:23:56 2023 -0600

    prueba if 2

commit bbbd7b4ea6287b85a7e3134dc448fc00ff9873df
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 19:13:01 2023 -0600

    prueba if1

commit d1fa7a756a55beea98cbfe5e1fc4562d82c2728d
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 19:04:29 2023 -0600

    prueba4 ajustes

commit 6cc73f40037761eb578560374e442bfdcbbab785
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 18:56:49 2023 -0600

    ajuste prueba3.txt

commit 10666deed9aff16f560bc5dbefdc4fb6124cc6f2
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 18:34:55 2023 -0600

    pruebas

commit d0cace51106f7cbf2e692cae1af733181d4625b9
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 18:11:19 2023 -0600

    factorial funciona

commit e5a3b12052277159112ca8bc4d3c899bda0c466a
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 18:04:06 2023 -0600
```

funciona el fibonacci

commit 779173756b06641d221f918f2f81eb8b3ac834ed
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 17:57:08 2023 -0600

funciones especiales jalan

commit b7fabd15444948fa148f8b72d8d34083c497e95f
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 16:24:47 2023 -0600

mat mult funciona

commit 1c5853a2c83e537699c9b266960f0b39844dac06
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 15:47:50 2023 -0600

mean ya funciona

commit 92ee7030b293d64c13c511e35e9939e7dac97272
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 15:41:21 2023 -0600

matrices jala pruebamatrix

commit bf8599bbaba76904c7c67eb719316eb407cf9119
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 15:04:34 2023 -0600

funciona pruebarray

commit 3f1b6a06fa71d106795356cc277b5e65b0f6118f
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 14:40:54 2023 -0600

gosub

commit b5b9940eb1f4afcac99b8a9753c5f719021e398e
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 14:36:40 2023 -0600

prueba3 ERA completado

commit 7e7919adccd23ad4e4c907226606d0a81c508cff
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat Jun 3 14:31:46 2023 -0600

```
    prueba4 al 100

commit 4a49c4ce5ef1d2e52b16df6c192a99cb91b9dfd3
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sat Jun 3 13:59:43 2023 -0600

    Exepcion ERA

commit 0823c1b3bad3b1722bea0ed3b875f355331b8c95
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 13:15:23 2023 -0600

    guarda valor de return en global

commit d36a79ab49de05a714e56e3b15e0302c61788da1
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sat Jun 3 13:01:19 2023 -0600

    jala prueba while2

commit 37608c5574437b8626ee0a421ee905f2f5512066
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 12:28:59 2023 -0600

    avances vm

commit c5dac3460973977f8b3dd9eeb0de8f85cec7c66d
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat Jun 3 11:13:07 2023 -0600

    prueba de ejecucion while1

commit 5a07d3f2abf55c17bc208e2203c93b9e52751600
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Fri Jun 2 23:41:35 2023 -0600

    elimine comentarios innecesarios

commit 953eb411c8ca832b37d213a7f629f7f317698542
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Fri Jun 2 23:40:50 2023 -0600

    ejemplo funcional

commit 54b095be2ed1fe5ff8fa90abe7c47f48887b064e
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Fri Jun 2 21:02:50 2023 -0600
```

ajustes

commit 889df1ad26d4d77648a462e4b6c4be95273b84d0
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Jun 2 20:46:12 2023 -0600

hashmap de valores asignados

commit fcc535193255317770d431cacd590de9bb1b3670
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Jun 2 18:46:58 2023 -0600

ajuste prueba

commit 47fb5b69ece9e276b03d748adfd1a88a5f3f622c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Jun 2 16:20:12 2023 -0600

ajustes parser

commit ba4242682b2dcbcc1e4d9ed593e8a5ae34d04bac
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Jun 2 16:10:53 2023 -0600

ajustes VM

commit b3c1c46e887b14dd9343c8741feb2c27fc1f89d3
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri Jun 2 12:51:01 2023 -0600

print terminado, error goto

commit d4f84a15d2ed323b1b9c6ee6252f805f942dd594
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Jun 2 10:49:18 2023 -0600

cambios

commit 56bb55f5f53c4429aa361b464ce9d6cd980d0c4d
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Thu Jun 1 20:45:07 2023 -0600

avances VM

commit a69ea31d3f1b592dbbf57cca2b9915d27ced48eb
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Thu Jun 1 18:12:49 2023 -0600

```
    elimine matrix.py que era para notas unicamente

commit cc572c6ab624fcab141606bab527ff93fe1cb2c6
Merge: 4e5c4bf ceb0bbb
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Thu Jun 1 14:12:23 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 4e5c4bf2b49cc27beab8d614d0e0c87b71051ea8
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Thu Jun 1 14:12:14 2023 -0600

    testing

commit ceb0bbbd7141132cfe9a0b7818cdb7401dd09cd2
Author: Beto11 <betotrevino@live.com.mx>
Date:   Thu Jun 1 09:11:03 2023 -0600

    VM: run, runQuads, operations, goto, gotoF, assign, print

commit 00f92a0879d77662b3b59ad7b60e1685f64e5178
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Wed May 31 20:42:23 2023 -0600

    agregue fibonacci

commit 524b181f8ccf3c86761dd71d34231809c412e1c7
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Wed May 31 20:22:48 2023 -0600

    ya estan todas las funcesp

commit d2b01ca3756814a7ac9090fd8e3d8d8f53f54d79
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Wed May 31 20:15:39 2023 -0600

    predict ya funciona en pruebafuncesp

commit 3cda25ce7198779d6a54da01559d037dfa189a23
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Wed May 31 19:55:06 2023 -0600

    ya funciona compile y genera cuadruplos

commit f85fad51cfa924b1d9f8d5a4c93cf10833bfb632
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
```

Date: Wed May 31 18:03:43 2023 -0600

    layers genera cuadruplos

commit 1dd12d1dc54df7762b834d867b022cf906614fdc
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 31 17:56:05 2023 -0600

    ya funcionan arrays con cuadruplos en pruebaarray.txt

commit 9f456c6b0deaf4d95a00aedc8bf9f87184cb5606
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 31 15:16:03 2023 -0600

    mean agregado a pruebamatrix

commit fa34b6c02199eb57b9842cca88eaca92fd2e4578
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 31 14:47:14 2023 -0600

    pruebamatrix funciona crea 2 matrix

commit 2b8e72e9f2ae6454a0e9f2eed581eae11f75b438
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 31 11:27:25 2023 -0600

    funciona declaracion de matrices con cuadruplos

commit df70359cc66335ba02f014151150f31beed36f19
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 30 22:56:27 2023 -0600

    avances array

commit 579e274dae8c5c42e51bd36cbbb680e4b6560559
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 30 17:46:43 2023 -0600

    ajuste

commit 3b0f7d5111e556445da8cdac976cdd18a2934078
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 30 17:45:51 2023 -0600

    implementacion inicial vm hecha por beto

commit 01d8e12f60848f8797c2874ed8a1eeccd782fe5c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>

Date: Tue May 30 15:13:12 2023 -0600

    elimine prints de testing en if

commit 3769501ddee538090eab81c9ca3975963abbd826
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 30 15:11:12 2023 -0600

    ya funciona while con pruebawhile

commit 2d2add263163da9af67cb4210816d4f1a7cf7458
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 30 13:14:22 2023 -0600

    funcionan AND y OR

commit bb02daade187519a52a85568be4737dc610049c0
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 30 11:44:29 2023 -0600

    cuadruplos de comparativos funcionando con tablas de variables y temporales

commit 8dbd451e5517a7af776b3fb21bbdb78982b0a99a
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 29 22:41:58 2023 -0600

    if else con temps

commit 19cf571a999871cd772171453d84b532ceffcaa5
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 29 18:19:24 2023 -0600

    implemente tabla de temporales con direccion de memoria

commit f274e529dfc88e5a56f10c96f119d443a0be023e
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 29 09:18:19 2023 -0600

    ajuste

commit be2a9982fb47c690efb0cce11da25f3ca5709d0b
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 29 08:49:24 2023 -0600

    tempcounter a getTemp

commit bc17c873e4621517ef7018eac74f0c161c30077f
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>

Date: Sun May 28 23:54:22 2023 -0600

    Update README.md

commit 3fe43ca215d364508a59f7c2e77db5a46623d9bf
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 28 23:52:53 2023 -0600

    debugging

commit 88e1d3be1fd45a32aec25eb63e7a6c9ce8d0f338
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 28 19:53:54 2023 -0600

    quadselse funciona

commit 3487f7bc960d743deb4919aab210235aebd8ae74
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 28 14:02:56 2023 -0600

    if sin else implementado

commit d26eb2215d859ac4dbaf33ea000728c311117e8c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 17:48:14 2023 -0600

    ajuste a prueba4

commit f33a4a4ee4160d8ca80db6ec4e12386eb0de2785
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 17:22:45 2023 -0600

    ajuste notas

commit 8106fe4c0036b7beb40db2973ed14d70b98d258d
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 17:21:18 2023 -0600

    testing

commit 9dd626ccb86b87d326fb959211c08694c928d0a3
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 17:09:18 2023 -0600

    cuadruplos al 100 de asignacion de llamadas de funciones normales y recursivas a variables

commit b0ca560996b46bc17fc7bc71ca69383e05cde37b

Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 16:51:40 2023 -0600

    ya almacena el valor del return de funciones tipo int y float

commit 11471e35e0c914e6d8baa241caa585e97e7f4289
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 16:04:00 2023 -0600

    debugging

commit f33b11312b5500ec7fa38305b8c316ac94e66e83
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 12:14:37 2023 -0600

    error cases cuando asignamos a variables que no existen

commit d712cc65514a2ebdad39e1f20aa83a1fef6ca7e3
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 11:37:02 2023 -0600

    implemente mainfunction

commit b767f7a31223032621e4bdd36a96fc903888256f
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 27 11:12:53 2023 -0600

    error cases y documentacion

commit d7def54cb18ca05f2266bf8fd0506db777f86cd5
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 22:20:12 2023 -0600

    comentario linea 391

commit 08e5ad5b793d2b1361d46d919525843c5b946c1e
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 22:16:44 2023 -0600

    ya esta al 100 la creacion de funciones

commit c48c9a68ae6c9f233a725b51910e794cdc859915
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 22:00:20 2023 -0600

    ya agrega tempcounter a cada func

commit 92960b575c559ffbc094b261675c60879eab9398

Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 19:37:01 2023 -0600

    tabla de funciones ya guarda correctamente startAddress

commit 175ebc918dfcb0c5927d9cace3bd2f4402e4061a
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 19:16:13 2023 -0600

    documentacion call

commit cf284f42f99d5dd0323eca470a4af8464a77ef96
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 19:13:22 2023 -0600

    agarra parametros en orden correcto y hace cuadruplos bien

commit 3c99fe29f46903c13a02f79eb25ce83ad3a90662
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 11:13:03 2023 -0600

    ya crea gosub con start address de cada funcion

commit 41062eb040fc06c79031d737b343854d00ddb2a0
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 26 11:05:21 2023 -0600

    funccall ya funciona con y sin parametros y genera cuadruplos

commit e9fee477ea860fe39a16541cb8aa45adb1ac98c2
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Thu May 25 23:13:57 2023 -0600

    print y read agregados y funcionales

commit ce614fbb3606576bd6f8e93be8372739f75f7960
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Thu May 25 19:39:54 2023 -0600

    corregi prueba3 y quite comentarios innecesarios

commit ecee1974282dbc12b5f90b62d5c7bcba8d1871e3
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Thu May 25 19:34:10 2023 -0600

    corregi orden de operaciones y ya funcionan varios parametros

commit 3bffacfaf546bcb0509f94eb23260a4cacc2d7c3

Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Thu May 25 13:01:15 2023 -0600

    ajuste a parametros, para que funciones puedan recibir muchos parametros

commit 9ffd7801bba9288db685170e63f5c952f34a94e0
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 24 18:09:38 2023 -0600

    cambios a cuadruplos operators

commit 8f30c28066351ce5aa77cd9d69d40aec63d865c6
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 24 13:15:11 2023 -0600

    goto al main cuadruplo agregado

commit 230fa3ae14dd69a1d7f27a2cf0461aab119e1467
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 24 10:16:42 2023 -0600

    cuadruplos funcionales

commit a9ca877b2576f9b4d220444b6a8c7c4c910d08d3
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 21:22:06 2023 -0600

    implemente tabla de constantes y ya se agregan cuando asignamos valores

commit 1d5f55569796ca167cc98e8957f996fbeccc646d
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 18:38:26 2023 -0600

    assign funcional

commit b9aaa2cd6607728921ed1d9e0d9517a39cf2cd08
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:34:57 2023 -0600

    valida que no se repitan funciones

commit 58b85688706fc2e0f5494fe8038ba7dcb24d4709
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:19:12 2023 -0600

    pycache

commit 43ae20d7b87b345714afdb69f3a0e0a47ef9cacc

Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:18:44 2023 -0600

    merge if y while

commit 29799cd110fb347b0d486b7ba059071308b1ee23
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:17:47 2023 -0600

    pycache

commit 422e0ae4b9e8543dd835f5a540a727d5ffd21e73
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:17:24 2023 -0600

    ajuste

commit a89e845fb9f65dbba79cd7380a024051d24910f7
Merge: cb2284c 5f4b66c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:16:20 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit cb2284cb9dd001bbfdd76ee3f1693e79e2c76697
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 23 12:15:32 2023 -0600

    debugging decfunc

commit 5f4b66c229596b399cca4a8f0e8cdcc3092a47f7
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 22 20:06:38 2023 -0600

    Update README.md

commit 72818d4a16d9a2e326d775ad476f7bc1d0d53f16
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 22 11:10:48 2023 -0600

    ajuste a lexer

commit ea526440c78e38debbb0adb8453359ac2919574e
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 22 11:00:25 2023 -0600

    lexer con estatutos de if y while

commit 8fcedf051d53e654b6fda3330370651465c34ec7
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 22 10:43:21 2023 -0600

    imports innecesarios eliminados

commit 1ba9e1265377d1bb5c08e3fd37d31925f96ed41c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 21 21:36:21 2023 -0600

    ajustes

commit aaa8080c434c1e6b14e50efb6a7e3b3e60590e79
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 21 20:35:10 2023 -0600

    ya funcionan funciones void

commit fe7dd2db69fc6c1fa8bf5a77fe3e81e146e095f9
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Thu May 18 17:44:26 2023 -0600

    ya funciona tabla de funciones, falta integrar temporales

commit 2d374330c07025f98a44fc89f7d90c9a0f3d63ed
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 17 23:35:18 2023 -0600

    se separaron tablas de variables globales de locales

commit 202129e3eddb059d7cbf5ddfdaea4ae813d98587
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 17 23:34:08 2023 -0600

    primera implementacion de tabla de variables locales en tabla de funciones

commit 2d260d1f793851f080f176c376fc8ed163d215d8
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Wed May 17 16:23:53 2023 -0600

    debugging y gexp

commit c89e71184e7bfc6500a559fe5bc02cfb45cf6a8d
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Wed May 17 16:23:26 2023 -0600

    ajustes gexp

commit c2eae1b1b443444b35e722428c22576a8d69a208
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 17 12:51:22 2023 -0600

    debugging assign

commit 865450c729bbf495b3d7e25bc5f92212364904bc
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Wed May 17 12:03:19 2023 -0600

    assign funciona completamente y corregi error handler de decvar

commit 177cf7f988e2fc00afa2b1aa47d78fd082e77db5
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 16 13:46:26 2023 -0600

    decvar ahora puede crear muchos tipos de variables/ es recursiva

commit d07edb5c14da98c3494c9db3a36e8aabbd8834c2
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 16 13:06:09 2023 -0600

    ya funciona add_varlocal

commit dc6a99b514eff2bfb392d9153d625929bff636c7
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 16 12:56:37 2023 -0600

    ya funciona add_var_global

commit 4ea7af53b97fd5e8437f647c0f19d40c511e4cc6
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 15 01:17:14 2023 -0600

    Update README.md

commit 78ccd8515609a547e4ab8495c24e7dd11c5316d9
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 01:09:44 2023 -0600

    borre comentarios que ya no sirven

commit c1bc37d41f21f56a6c0780382a54056a53c81dfc
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 01:08:04 2023 -0600

    correccion generador de cuadruplos y pila cuadruplos

```
commit 2cad3d85f2388b432fa2cd95154fb42c044fd1ca
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Mon May 15 00:38:06 2023 -0600

    cambios del beto

commit dff7cba32e3ef89d097086d471c6a91f367b1b71
Merge: 6d3692a 3f9b9ad
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Mon May 15 00:28:14 2023 -0600

    merge

commit 6d3692a0ddcf39ea98ad7033e3bdeb94f7705304
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Mon May 15 00:26:00 2023 -0600

    decvar funcionando al 100

commit 3f9b9ad94078b4a0ea1c0f980241a6a88b4b0b4c
Author: Beto11 <betotrevino@live.com.mx>
Date:   Sun May 14 19:33:01 2023 -0600

    se mejoro func while e if

commit a1296c43df786fb144b710620ab355c21e2a5dd5
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun May 14 17:24:38 2023 -0600

    borre linea innecesaria

commit 6ed4a6a46d08fda4ca8c21addc6963dbfa7c16fc
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun May 14 17:17:05 2023 -0600

    ajustes

commit 7dfecfcbe0e48ff1af0953d25c4e9c6a493e7c87
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun May 14 16:58:25 2023 -0600

    ya podemos guardar variables

commit bbad786bb7905ce699870904ecbe449026f325e3
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun May 14 16:46:26 2023 -0600

    cambios a decvar, type, exp y program para debugging
```

```
commit d19b09c056b32469cb123b54aa43fb34fb40d342
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat May 13 14:02:12 2023 -0600

    correcciones a p_program y prueba funcionando

commit 487dc0cdf3b93ea5b76136b656c13d4763c9ab52
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat May 13 11:53:47 2023 -0600

    testing y debugging de program y decvar

commit 738bde75894e2b62218abf51937b6a8c1190f2fd
Merge: e17d9ea fc2e703
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat May 13 10:54:49 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit e17d9ea6a209fdee9b5db4e80df4b8d0b8d35731
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sat May 13 10:54:46 2023 -0600

    merge

commit fc2e703f33d50b314d29013c5b8cfe5ed39e5ec3
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sat May 13 02:04:26 2023 -0600

    se comento los quads while if

commit dd042dadd104250ba292f797eb267f082e11d824
Merge: 99178c3 30d35df
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sat May 13 01:56:19 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 99178c34ea747577e07435d245ac61d8df2df133
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sat May 13 01:56:00 2023 -0600

    se implemento los quads para while e if, solamete falta checar si funcionan

commit 30d35df03dd1bd1cbcae4e15632a479a6290d456
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Fri May 12 17:49:09 2023 -0600
```

```
    borre comentario de problema solucionado

commit d1c0c520f5d4646d2b0f54120495672e5fbb2198
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Fri May 12 17:48:08 2023 -0600

    debugging

commit d6ae2481df8a4dfb60e1e2be65e8874361a53b5c
Merge: a48999b c18aeca
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date:   Fri May 12 13:11:45 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit a48999b303a4d4ef4ba488b525bb30c0620b95b1
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date:   Fri May 12 13:11:38 2023 -0600

    implementación ejemplo tabla de funciones

commit 8c7111a7d2aab30d403170569e445b0189c37567
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date:   Fri May 12 13:02:32 2023 -0600

    tabla de funciones actualizada

commit c18aeca03ee7dfd23330570f28e136e18ffe96a8
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Fri May 12 00:21:45 2023 -0600

    checaron cambios

commit 67619421f0b2b307b8dd5cd8235d9860d683bafa
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Fri May 12 00:20:00 2023 -0600

    se agrego p_error, y checar si la implementacion del gen codigo es correcta

commit d8dcf8d0feacfda137b618989c40cbea6a5b042b
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Fri May 12 00:01:40 2023 -0600

    se corrigio el print, se agrego el body, y se arreglaron algunos errores de recursion infinita

commit f45d97f6140f3450f375e3e52ab8bc80f43f7a37
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
```

Date: Thu May 11 10:06:40 2023 -0600

    separe function en 2: funcion normal y funcion void

commit f3bef5b8b41112757e069a4a519f8f0a27c7a984
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Thu May 11 09:56:35 2023 -0600

    debugging

commit bf706f98a39440760e9e9560419bd238c2344e0c
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Wed May 10 16:55:03 2023 -0600

    debugging y archivo de prueba

commit dac364f9f453bd471ae491d1bd693b663cb85e47
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 9 14:24:55 2023 -0600

    cambios

commit 8843c43c9b29e57aa8037fda83c149af37477e2a
Merge: 01d1af7 9632594
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 9 13:31:20 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 01d1af79db74378b204069cd093be2193a99609f
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 9 13:30:53 2023 -0600

    implementación de tabla de variables y directorio de procedimientos, ademas de rangos de memoria

commit 96325942e174857ceccd63d5092ec3ab1eab2b85
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Sun May 7 23:15:49 2023 -0600

    Update README.md

commit d070026df198331c45f20610e861c93862de039f
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri May 5 13:48:09 2023 -0600

    se arreglo el lexer y parser

commit 63dd632b11a6186ff3100221bdadbf3670c8b368
Merge: 67a3ee7 3c600fd
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Apr 28 17:49:40 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 67a3ee7949d024094e7df101115682120b45e47c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Apr 28 17:49:14 2023 -0600

    testing cubo semantico

commit 3c600fd2c875101d394b41c74db0f5ff35c0e2c3
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Fri Apr 28 17:38:16 2023 -0600

    Update README.md

commit bbc5ffb45cbc4586a0d6165668302ef5a162257e
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Apr 28 17:36:29 2023 -0600

    cubo semántico primera implementación

commit ff165363a530d0a801e2ebcade23856d653fd263
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri Apr 28 13:19:40 2023 -0600

    Parser completado, checar detalles

commit eee18c5e409020cbe267c81b9e00a6ca4a76d02a
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Tue Apr 25 13:43:12 2023 -0600

    Update bixoLexer.py

commit 3cbd23e89b240d6b8958cbd5393ca74cff76b835
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Sun Apr 16 22:33:51 2023 -0600
Date: Sun Apr 16 22:33:51 2023 -0600

    Update README.md

commit 220bf0c37446bcf4caed5d3be2f2a433e0ae1b97
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 22:30:51 2023 -0600

Parser iniciado y lexer implementado

commit e90cbc0cefc48cd12e8558efe86fb4cad8c4c6be
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:57:16 2023 -0600

debug

commit b64cf762dfad6ba3d1a84e25e7db2e449f4dc92c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:53:03 2023 -0600

lexer implementado

commit 0b7186575bec79a3891092b874d4a3cf3e8a6a8a
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:45:26 2023 -0600

implementacion inicial parser

commit 8dcd0c65e199fa4ac0d227a300993c0d133d4354
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:23:59 2023 -0600

lexer y parser separados

commit a846c6532a769810510ef7b9dd237336d6213bd8
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:03:46 2023 -0600

creación de archivo de parser y lexer

commit ad3513ccc8cf0ccdc81dac465acf70c82659b1a9
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Sun Apr 16 21:01:23 2023 -0600

Initial commit

commit 3a7783e386f85958c9ac4d0b248adfbeb3677046
Author: Beto11 <betotrevino@live.com.mx>
Date: Mon May 22 10:44:49 2023 -0600

while

commit da8f5f829671564fa8f096870c76399d42cea68a
Author: Beto11 <betotrevino@live.com.mx>
Date: Mon May 22 07:53:24 2023 -0600

```
    while terminado falta testing

commit 9c848d42ebad60417d80bcd920e7e14f452ee202
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun May 21 21:31:49 2023 -0600

    if else funcional

commit 3d1a8314abf997c57ac4eaea442c20b88498f202
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sun May 21 20:37:39 2023 -0600

    lineas comentadas

commit 29bfcad133efbb8d0ef9cd181446ef714f058e23
:...skipping...
commit d1603e17d7c14b8fb81477da1e6413a82f51d4cb (HEAD -> beto, origin/beto)
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date:   Mon May 22 11:23:49 2023 -0600

    merge con main

commit 3a7783e386f85958c9ac4d0b248adfbeb3677046
Author: Beto11 <betotrevino@live.com.mx>
Date:   Mon May 22 10:44:49 2023 -0600

    while

commit da8f5f829671564fa8f096870c76399d42cea68a
Author: Beto11 <betotrevino@live.com.mx>
Date:   Mon May 22 07:53:24 2023 -0600

    while terminado falta testing

commit 9c848d42ebad60417d80bcd920e7e14f452ee202
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date:   Sun May 21 21:31:49 2023 -0600

    if else funcional

commit 3d1a8314abf997c57ac4eaea442c20b88498f202
Author: Alberto Treviño <betotrevino@live.com.mx>
Date:   Sun May 21 20:37:39 2023 -0600

    lineas comentadas

commit 29bfcad133efbb8d0ef9cd181446ef714f058e23
```

Author: Beto11 <betotrevino@live.com.mx>
Date: Sat May 20 11:31:10 2023 -0600

    fill con errores, e implementacion del WHILE

commit 3f686fcfd76f38e8a4e06aa5f2cdb65e66b36241
Author: Beto11 <betotrevino@live.com.mx>
Date: Fri May 19 12:10:09 2023 -0600

    no le di save, ahora si temporales bien

commit 58bb600a71bb92564e4af0b7e64cc9b2892defd7
Author: Beto11 <betotrevino@live.com.mx>
Date: Fri May 19 12:09:31 2023 -0600

    se agrego los temporales

commit a84396b04679c7b21e6bf3af22c989771fd99fc4
Author: Beto11 <betotrevino@live.com.mx>
Date: Fri May 19 09:42:53 2023 -0600

    Ajustes IF (ToF jumps, counter)

commit 877544e86bc18d1705e26ec515fd778b5fed3eea
Author: Beto11 <betotrevino@live.com.mx>
Date: Thu May 18 10:11:52 2023 -0600

    cambio del gotoF

commit b3e56122c8c2e1e09065eb81f4cff31b0156bf6d
Author: Beto11 <betotrevino@live.com.mx>
Date: Thu May 18 08:46:17 2023 -0600

    if casi terminado junto con else

commit cc6e11586fdb28c83964f04ac8c29b34fd145251
Author: Beto11 <betotrevino@live.com.mx>
Date: Wed May 17 16:07:55 2023 -0600

    stacks funcionales de operandos y operadores IF

commit b2a448cd6d7fbe8dfdd1229da55544adcdb88c7f
Author: Beto11 <betotrevino@live.com.mx>
Date: Wed May 17 15:29:09 2023 -0600

    merge

commit c18e495047d488c8ec24463fdbd2e98bce1de40d

Author: Beto11 <betotrevino@live.com.mx>
Date: Wed May 17 15:27:36 2023 -0600

    ajustes

commit 7291da60d2af8374c9e17ed7a4fb00fd3142d6ab
Author: Beto11 <betotrevino@live.com.mx>
Date: Wed May 17 15:25:27 2023 -0600

    merge con main

commit 2b03d6b2547b9e41ea36f9d765ea087f2f0c10ea
Author: Beto11 <betotrevino@live.com.mx>
Date: Wed May 17 15:17:29 2023 -0600

    cambios if, param, !=

commit 15e2a56f24799fbbe06b5a964c662df7e213b3cc
Author: Beto11 <betotrevino@live.com.mx>
Date: Tue May 16 18:39:51 2023 -0600

    IF mejorado

commit 2285d9eb99d01fc4b277e3329ca9acd565ca9945
Author: Beto11 <betotrevino@live.com.mx>
Date: Tue May 16 15:23:25 2023 -0600

    se corrigio el param en parser

commit 47eed6535d1223c97dc872c199dec56dbbee05d3
Author: Beto11 <betotrevino@live.com.mx>
Date: Tue May 16 14:50:21 2023 -0600

    cambios if

commit 177cf7f988e2fc00afa2b1aa47d78fd082e77db5
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 16 13:46:26 2023 -0600

    decvar ahora puede crear muchos tipos de variables/ es recursiva

commit d07edb5c14da98c3494c9db3a36e8aabbd8834c2
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 16 13:06:09 2023 -0600

    ya funciona add_varlocal

commit dc6a99b514eff2bfb392d9153d625929bff636c7

Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 16 12:56:37 2023 -0600

    ya funciona add_var_global

commit 4ea7af53b97fd5e8437f647c0f19d40c511e4cc6
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Mon May 15 01:17:14 2023 -0600

    Update README.md

commit 78ccd8515609a547e4ab8495c24e7dd11c5316d9
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 01:09:44 2023 -0600

    borre comentarios que ya no sirven

commit c1bc37d41f21f56a6c0780382a54056a53c81dfc
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 01:08:04 2023 -0600

    correccion generador de cuadruplos y pila cuadruplos

commit 2cad3d85f2388b432fa2cd95154fb42c044fd1ca
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 00:38:06 2023 -0600

    cambios del beto

commit dff7cba32e3ef89d097086d471c6a91f367b1b71
Merge: 6d3692a 3f9b9ad
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 00:28:14 2023 -0600

    merge

commit 6d3692a0ddcf39ea98ad7033e3bdeb94f7705304
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Mon May 15 00:26:00 2023 -0600

    decvar funcionando al 100

commit 3f9b9ad94078b4a0ea1c0f980241a6a88b4b0b4c
Author: Beto11 <betotrevino@live.com.mx>
Date: Sun May 14 19:33:01 2023 -0600

    se mejoro func while e if

commit a1296c43df786fb144b710620ab355c21e2a5dd5
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 14 17:24:38 2023 -0600

    borre linea innecesaria

commit 6ed4a6a46d08fda4ca8c21addc6963dbfa7c16fc
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 14 17:17:05 2023 -0600

    ajustes

commit 7dfecfcbe0e48ff1af0953d25c4e9c6a493e7c87
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 14 16:58:25 2023 -0600

    ya podemos guardar variables

commit bbad786bb7905ce699870904ecbe449026f325e3
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun May 14 16:46:26 2023 -0600

    cambios a decvar, type, exp y program para debugging

commit d19b09c056b32469cb123b54aa43fb34fb40d342
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 13 14:02:12 2023 -0600

    correcciones a p_program y prueba funcionando

commit 487dc0cdf3b93ea5b76136b656c13d4763c9ab52
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 13 11:53:47 2023 -0600

    testing y debugging de program y decvar

commit 738bde75894e2b62218abf51937b6a8c1190f2fd
Merge: e17d9ea fc2e703
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 13 10:54:49 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit e17d9ea6a209fdee9b5db4e80df4b8d0b8d35731
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sat May 13 10:54:46 2023 -0600

    merge

commit fc2e703f33d50b314d29013c5b8cfe5ed39e5ec3
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Sat May 13 02:04:26 2023 -0600

    se comento los quads while if

commit dd042dadd104250ba292f797eb267f082e11d824
Merge: 99178c3 30d35df
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Sat May 13 01:56:19 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 99178c34ea747577e07435d245ac61d8df2df133
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Sat May 13 01:56:00 2023 -0600

    se implemento los quads para while e if, solamete falta checar si funcionan

commit 30d35df03dd1bd1cbcae4e15632a479a6290d456
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 12 17:49:09 2023 -0600

    borre comentario de problema solucionado

commit d1c0c520f5d4646d2b0f54120495672e5fbb2198
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri May 12 17:48:08 2023 -0600

    debugging

commit d6ae2481df8a4dfb60e1e2be65e8874361a53b5c
Merge: a48999b c18aeca
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Fri May 12 13:11:45 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit a48999b303a4d4ef4ba488b525bb30c0620b95b1
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Fri May 12 13:11:38 2023 -0600

    implementación ejemplo tabla de funciones

commit 8c7111a7d2aab30d403170569e445b0189c37567
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Fri May 12 13:02:32 2023 -0600

    tabla de funciones actualizada

commit c18aeca03ee7dfd23330570f28e136e18ffe96a8
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri May 12 00:21:45 2023 -0600

    checaron cambios

commit 67619421f0b2b307b8dd5cd8235d9860d683bafa
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri May 12 00:20:00 2023 -0600

    se agrego p_error, y checar si la implementacion del gen codigo es correcta

commit d8dcf8d0feacfda137b618989c40cbea6a5b042b
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri May 12 00:01:40 2023 -0600

    se corrigio el print, se agrego el body, y se arreglaron algunos errores de recursion infinita

commit f45d97f6140f3450f375e3e52ab8bc80f43f7a37
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Thu May 11 10:06:40 2023 -0600

    separe function en 2: funcion normal y funcion void

commit f3bef5b8b41112757e069a4a519f8f0a27c7a984
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Thu May 11 09:56:35 2023 -0600

    debugging

commit bf706f98a39440760e9e9560419bd238c2344e0c
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Wed May 10 16:55:03 2023 -0600

    debugging y archivo de prueba

commit dac364f9f453bd471ae491d1bd693b663cb85e47
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 9 14:24:55 2023 -0600

    cambios

commit 8843c43c9b29e57aa8037fda83c149af37477e2a
Merge: 01d1af7 9632594
Author: Juan Camilo Granados <juanca_grave@hotmail.com>

Date: Tue May 9 13:31:20 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 01d1af79db74378b204069cd093be2193a99609f
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Tue May 9 13:30:53 2023 -0600

    implementación de tabla de variables y directorio de procedimientos, ademas de rangos de memoria

commit 96325942e174857ceccd63d5092ec3ab1eab2b85
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Sun May 7 23:15:49 2023 -0600

    Update README.md

commit d070026df198331c45f20610e861c93862de039f
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri May 5 13:48:09 2023 -0600

    se arreglo el lexer y parser

commit 63dd632b11a6186ff3100221bdadbf3670c8b368
Merge: 67a3ee7 3c600fd
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Apr 28 17:49:40 2023 -0600

    Merge branch 'main' of https://github.com/JCGranadosV/BIXO

commit 67a3ee7949d024094e7df101115682120b45e47c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Apr 28 17:49:14 2023 -0600

    testing cubo semantico

commit 3c600fd2c875101d394b41c74db0f5ff35c0e2c3
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Fri Apr 28 17:38:16 2023 -0600

    Update README.md

commit bbc5ffb45cbc4586a0d6165668302ef5a162257e
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Fri Apr 28 17:36:29 2023 -0600

    cubo semántico primera implementación

commit ff165363a530d0a801e2ebcade23856d653fd263
Author: Alberto Treviño <betotrevino@live.com.mx>
Date: Fri Apr 28 13:19:40 2023 -0600

    Parser completado, checar detalles

commit eee18c5e409020cbe267c81b9e00a6ca4a76d02a
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Tue Apr 25 13:43:12 2023 -0600

    Update bixoLexer.py

commit 3cbd23e89b240d6b8958cbd5393ca74cff76b835
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Sun Apr 16 22:33:51 2023 -0600

    Update README.md

commit 220bf0c37446bcf4caed5d3be2f2a433e0ae1b97
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 22:30:51 2023 -0600

    Parser iniciado y lexer implementado

commit e90cbc0cefc48cd12e8558efe86fb4cad8c4c6be
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:57:16 2023 -0600

    debug

commit b64cf762dfad6ba3d1a84e25e7db2e449f4dc92c
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:53:03 2023 -0600

    lexer implementado

commit 0b7186575bec79a3891092b874d4a3cf3e8a6a8a
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:45:26 2023 -0600

    implementacion inicial parser

commit 8dcd0c65e199fa4ac0d227a300993c0d133d4354
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:23:59 2023 -0600

    lexer y parser separados

commit a846c6532a769810510ef7b9dd237336d6213bd8
Author: Juan Camilo Granados <juanca_grave@hotmail.com>
Date: Sun Apr 16 21:03:46 2023 -0600

    creación de archivo de parser y lexer

commit ad3513ccc8cf0ccdc81dac465acf70c82659b1a9
Author: Juan Camilo <70963790+JCGranadosV@users.noreply.github.com>
Date: Sun Apr 16 21:01:23 2023 -0600

Reflexión

Camilo:
El proceso de programar un compilador te obliga a esforzarte y salirte de tu zona de confort para hacer que el código sea funcional, entendiendo el funcionamiento, mediante un camino bastante desgastante debido a los obstáculos que se presentaron, ya sea por bugs, o alguna falla sintáctica o léxica. Uno de los mayores retos que tuvimos fue el de la deuda técnica, la cual nos llegó a afectar hasta las etapas más tardías de nuestro proyecto. Sin embargo es un proyecto muy llenador al completarlo debido a la satisfacción que brinda ver el código funcional, aceptando los casos de prueba planeados, ver su generación de tablas, cuadruplos, etc.

El aprendizaje de un compilador es un desafío que pone a prueba tu disciplina y conocimientos acerca de la programación y te enseña a tener una práctica constante y tener un mejor entendimiento de cómo se lleva a cabo las partes específicas del código.

Alberto:
Es un proyecto el cual requiere disciplina y esfuerzo dado a la complejidad que tiene, pone en práctica todos los conocimientos que llevemos a través de la carrera y perfecciona los detalles que nos faltaba pulir debido a la constante práctica para hacer que el código sea funcional. Nos ayuda a entender mejor el funcionamiento de un compilador e incluso de las funciones que utilizamos dia a dia ya que normalizamos su funcionamiento sin pensar. Y nos sirve como un buen proyecto para después comenzar la vida profesional.

Se puso en práctica varios aspectos importantes, tal como debuggear y entender el funcionamiento del código, ya que había casos en los que ocupamos del print para ir siguiendo la estructura de tal y ver en donde se presentaba el error y detalles similares.

Descripción del Lenguaje

Bixo es un lenguaje de programación en el cual se pueden desarrollar algoritmos sencillos junto con las operaciones básicas. Machine learning, ciclos, declaración de variables, funciones, tabla de funciones, arrays, matrices, multiplicación de matrices, funciones especiales de machine learning para hacer predicciones utilizando un modelo de tensor flow.

Bixo maneja errores de programación tal como: Índices fuera de rango, variables sin definir, errores de gramática(falta ;, etc), Rangos incoherentes dentro del array

Bixo tiene algunas limitaciones a la hora de traducir los cuádruplos en la máquina virtual principalmente en la llamada a funciones, lo cual complica las ejecuciones recursivas. No acepta valores negativos como parámetros en las funciones definidas. No puede declarar variables con valor asignado, sin embargo puedes declarar una variable y después asignarle un valor.

Descripción del Compilador

Equipo, lenguaje y utilerías

El compilador Bixo fue programado en Python utilizando la librería PLY para su analizador léxico y sintáctico. Se utilizó la librería Numpy y Tensorflow para su implementación de Machine Learning. Fue desarrollado en una Lenovo, 8GB de RAM con un procesador intel i5 10gen.y una Asus ROG Intel i7 con 16GB de RAM y una tarjeta gráfica GTX 1080, Utilizando VS Code como el IDE.

Descripción del Analizador Léxico

La librería PLY es leído por un parser. Los datos introducidos en el analizador son un flujo de tokens, generados por el analizador léxico. El final de una línea lógica está representado por el token NEWLINE Una línea física es una secuencia de caracteres terminada por una secuencia de final de línea.
Se utilizaron las siguientes palabras reservadas:
'program' : 'PROGRAM',
    'end' : 'END',

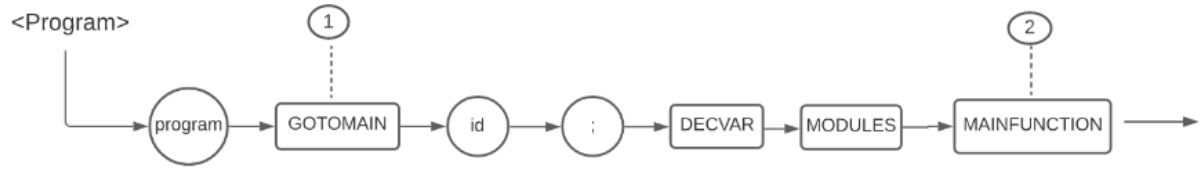
```
'main' : 'MAIN',
'var' : 'VAR',
'type' : 'TYPE',
'function' : 'FUNCTION',
'read' : 'READ',
'call' : 'CALL',
'if': 'IF',
'else': 'ELSE',
'while': 'WHILE',
'int': 'INT',
'float': 'FLOAT',
'bool': 'BOOL',
'string': 'STRING',
'true': 'TRUE',
'false': 'FALSE',
'print' : 'PRINT',
'void' : 'VOID',
'epochs' : 'EPOCHS',
'verbose' : 'VERBOSE'
```

Descripción del Analizador Sintactico

El analizador sintáctico obtiene una cadena de tokens del analizador léxico y verifica que dicha cadena pueda generarse con la gramática para el lenguaje fuente y lo que no esté definido/vacío se interpreta como epsilon.

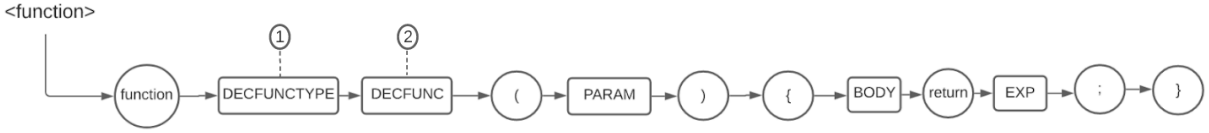
Descripción de Generación de Código Intermedio y Análisis Semántico

Program



1. Genera el cuádruplo “goto main”
2. Inserta el goto al main

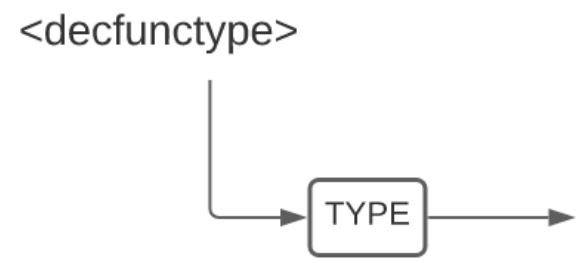
Function



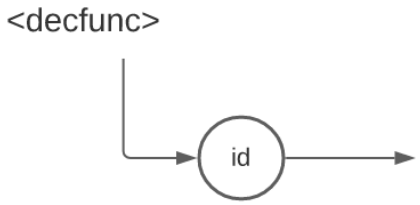
1. Define el tipo de función y lo almacena en las variables globales

2. Define el nombre de la función, crea el stack de funciones, y reinicia para futuras ocasiones.

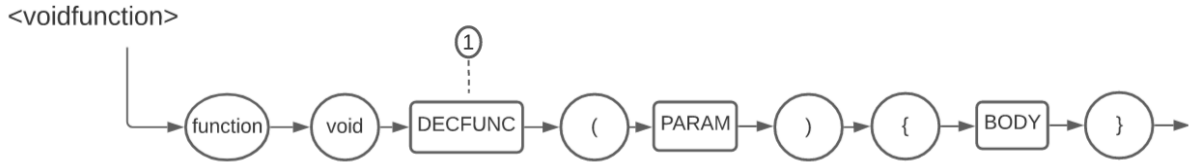
DecFuncType



DecFunc

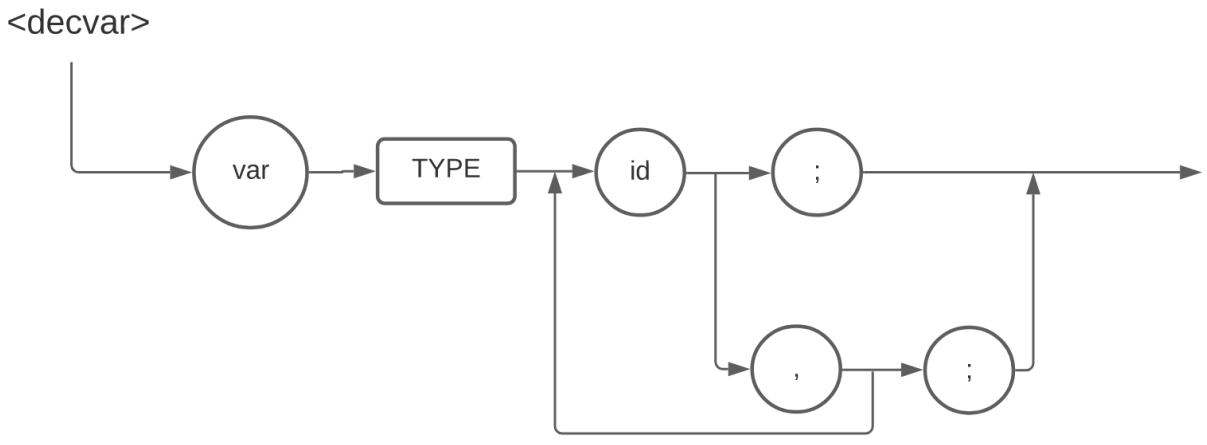


VoidFunction



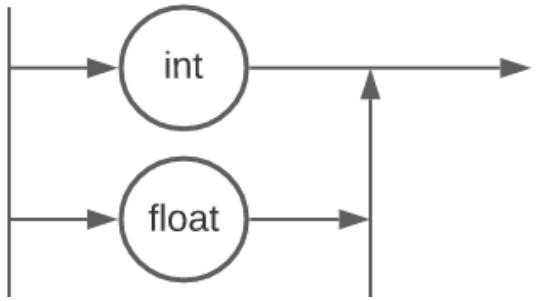
1. Define el nombre de la función, crea el stack de funciones, y reinicia para futuras ocasiones.

Dec-Var



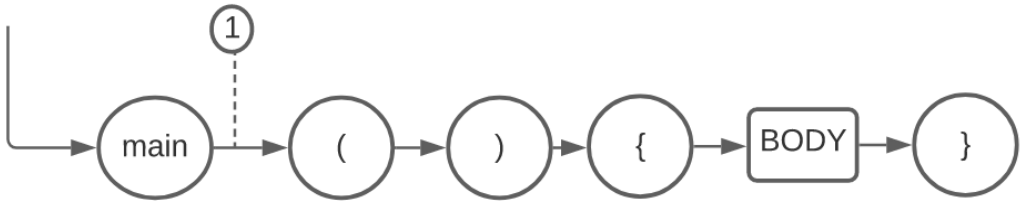
Type

<type>



MainFunction

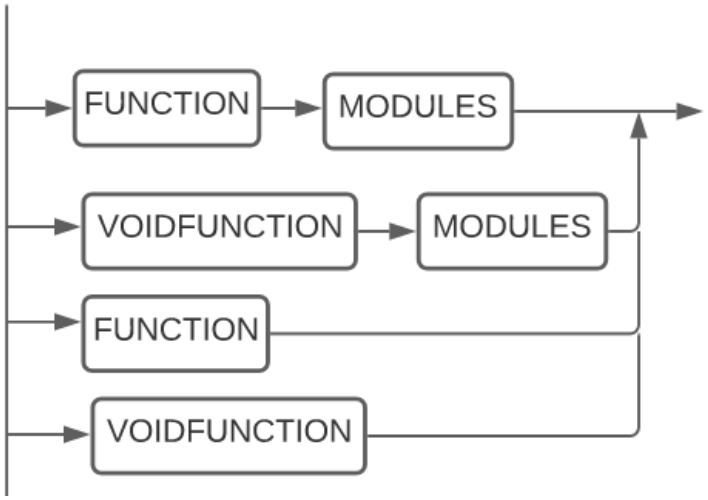
<mainfunction>



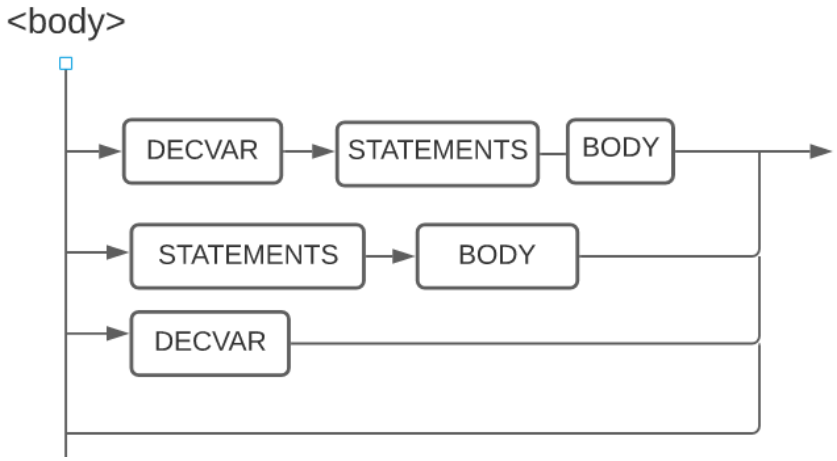
1. Rellena el primer cuádruplo con la localización del main

Modules

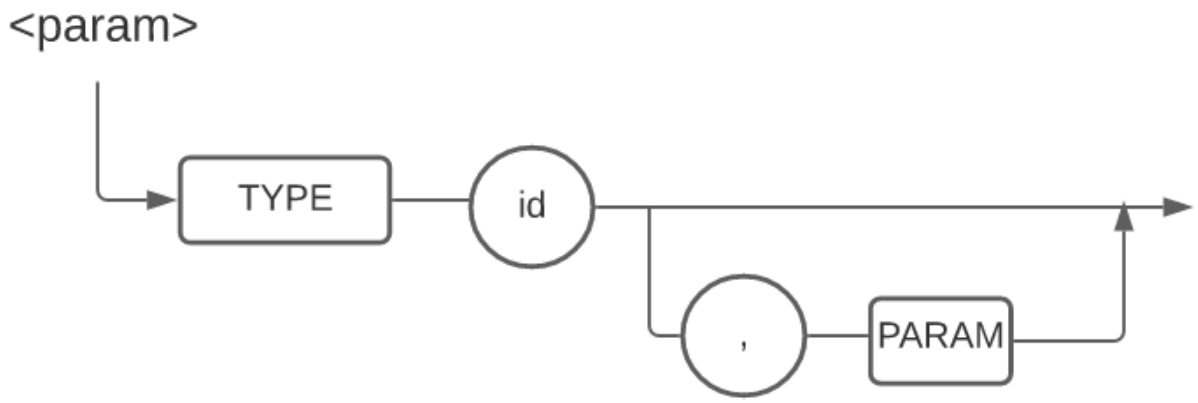
<modules>



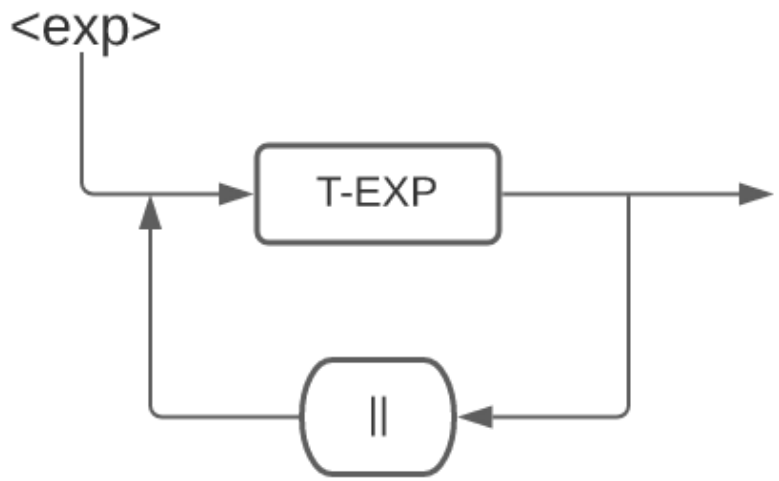
Body

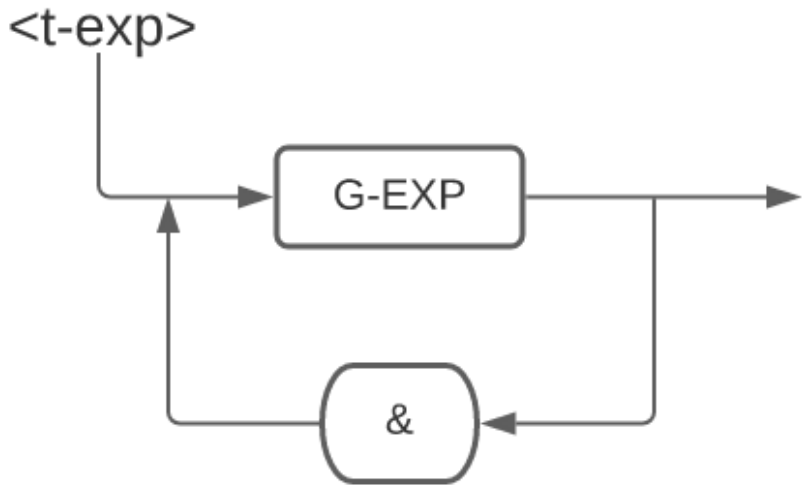


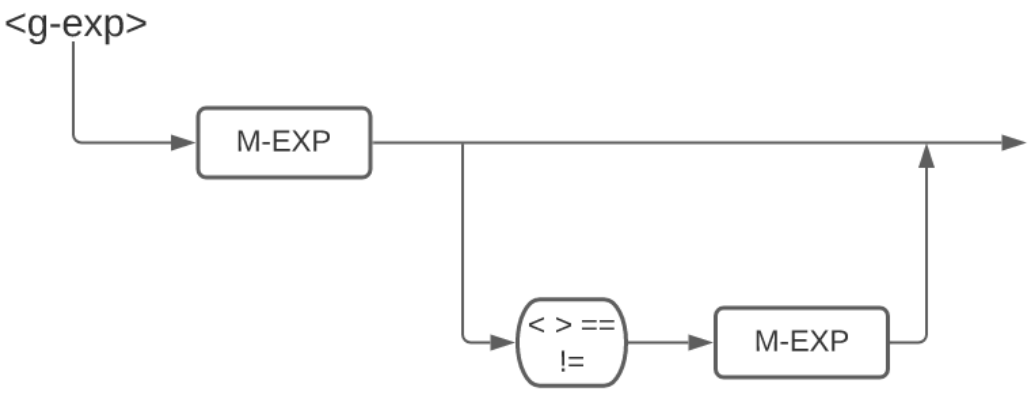
Parameters

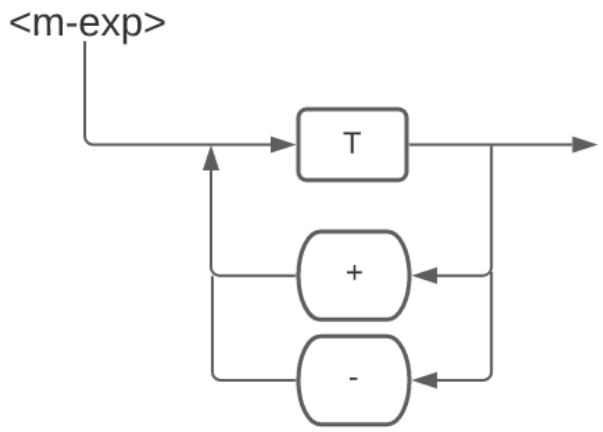


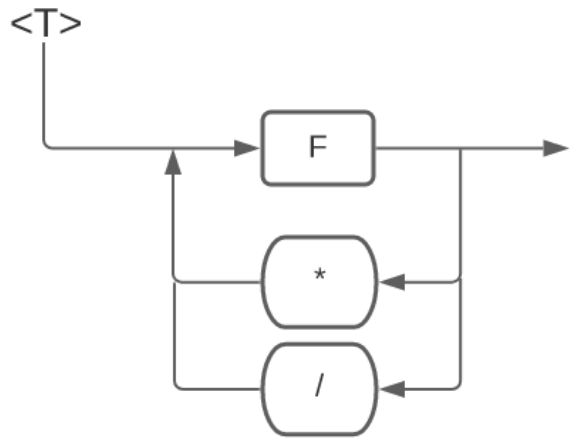
Expressions



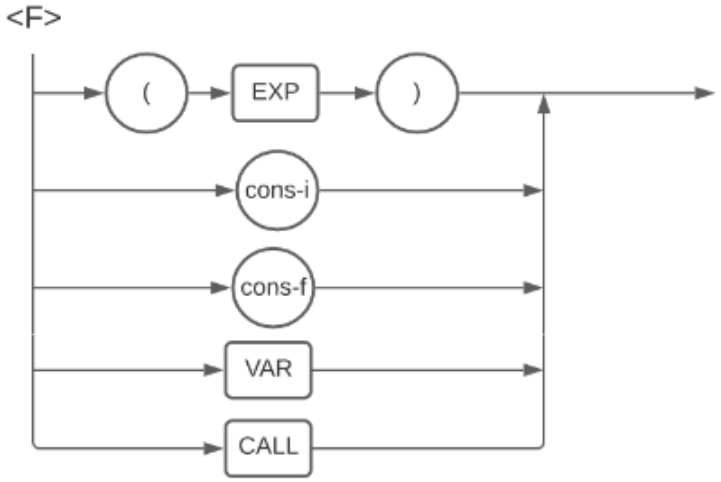

<t-exp>
G-EXP
&


<g-exp>
M-EXP
< > ==
!=
M-EXP

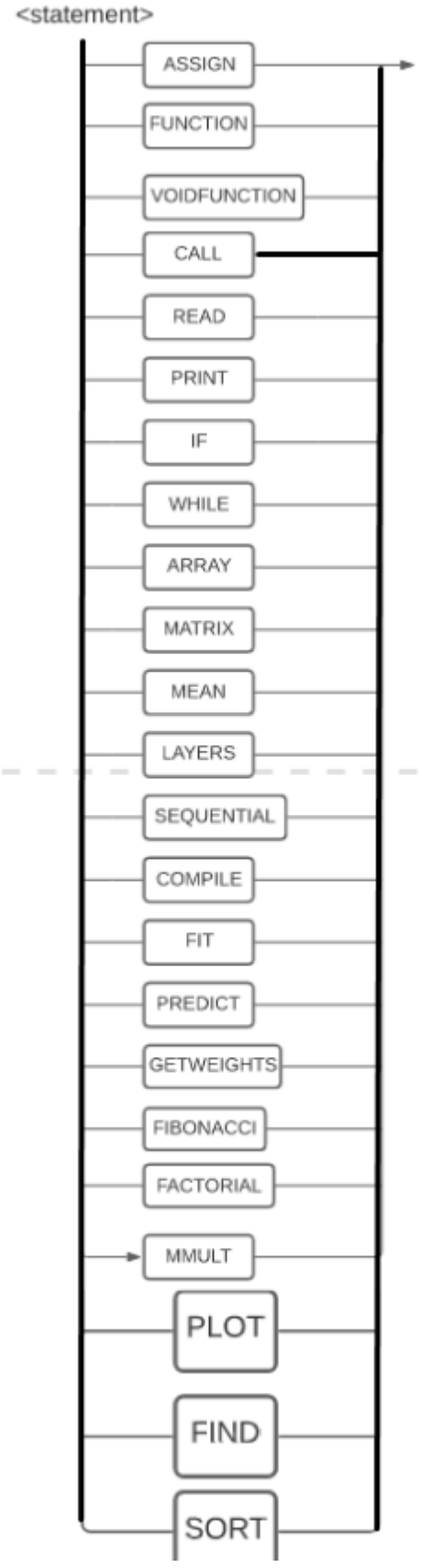

<m-exp>
T
+
-



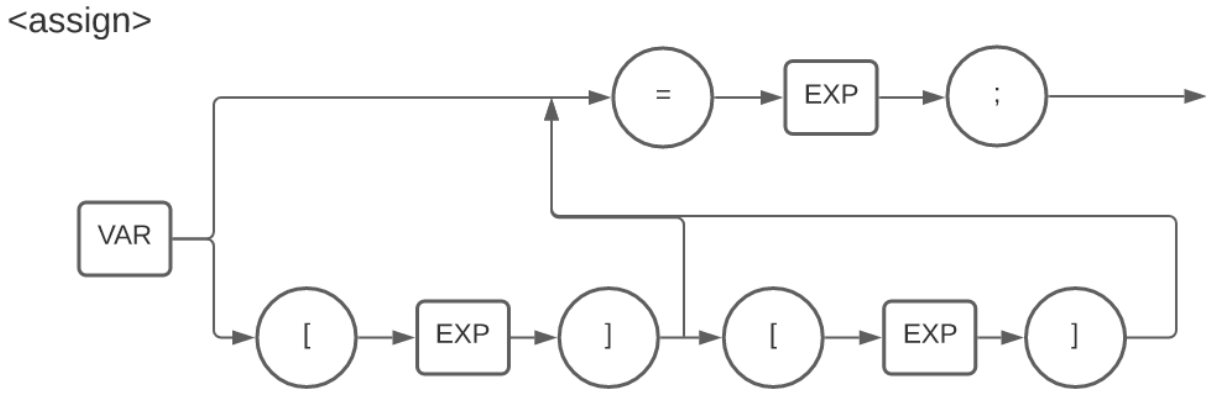
Factor



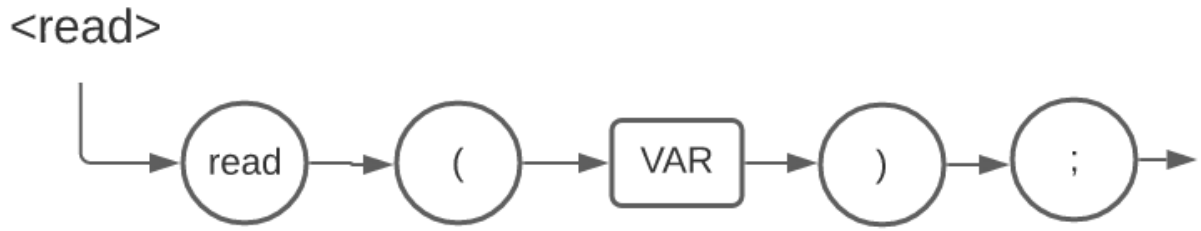
Statement


<statement>
ASSIGN
FUNCTION
VOIDFUNCTION
CALL
READ
PRINT
IF
WHILE
ARRAY
MATRIX
MEAN
LAYERS
SEQUENTIAL
COMPILE
FIT
PREDICT
GETWEIGHTS
FIBONACCI
FACTORIAL
MMULT
PLOT
FIND
SORT

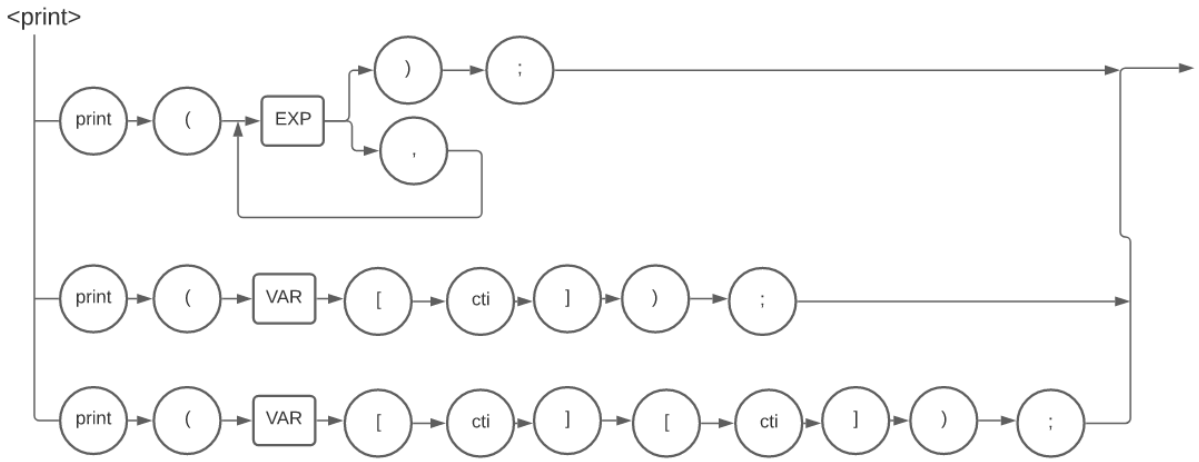
Assign



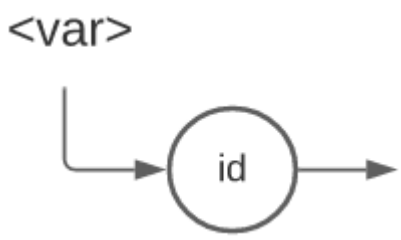
Read



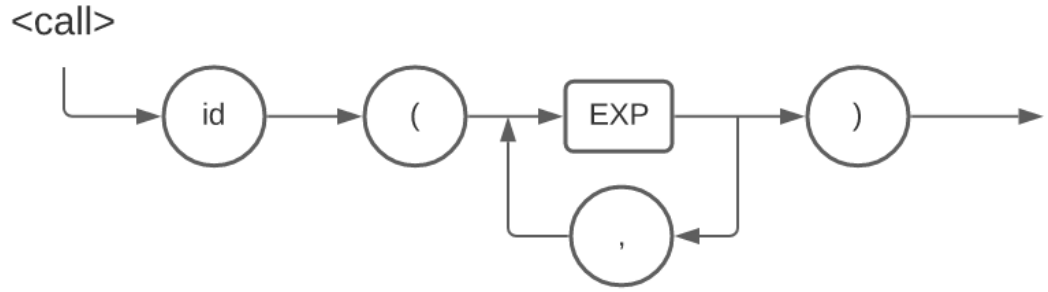
Print



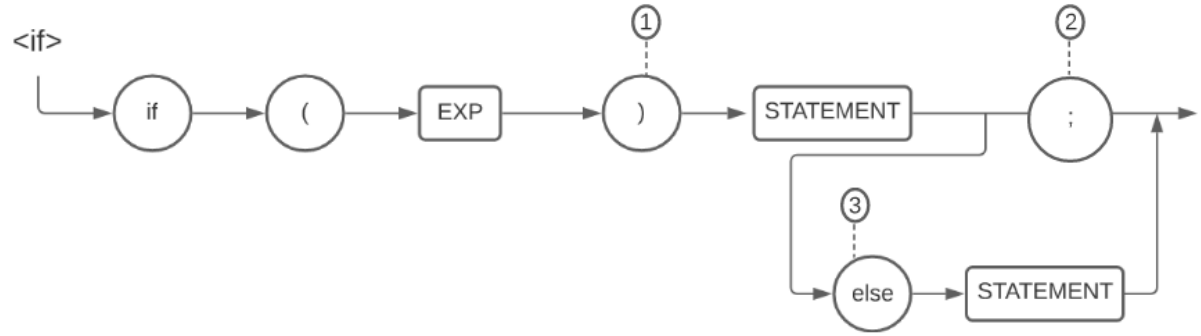
Variables



Call

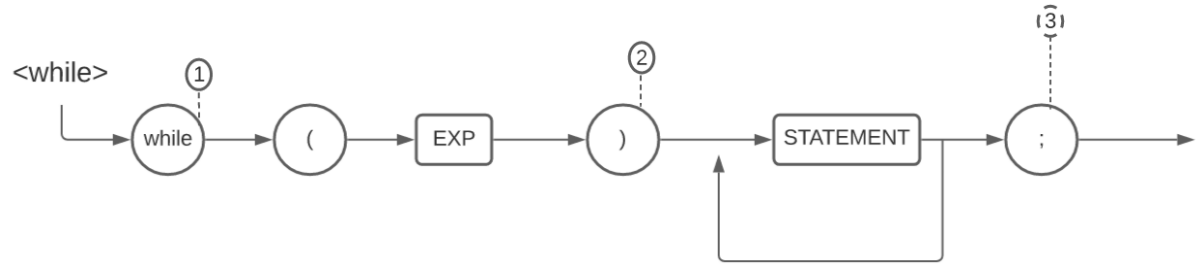


If



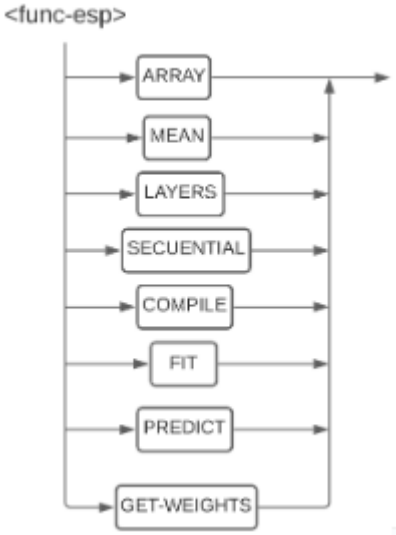
1. Hace pop, verifica si el type es booleano y genera el cuádruplo gotoF
2. Hace pop de lo pendiente, y finalmente hace fill al gotoF
3. genera el cuádruplo goto, en caso de ser false hace pop, y hace fill al goto.

While

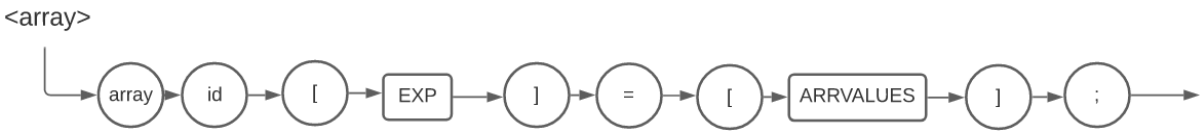


1. Hace pop del contador de saltos
2. Hace pop a la tabla de experiencias, genera el cuádruplo gotoF
3. Hace pop del stack de los saltos, se genera el goto, y hace fill al contenido

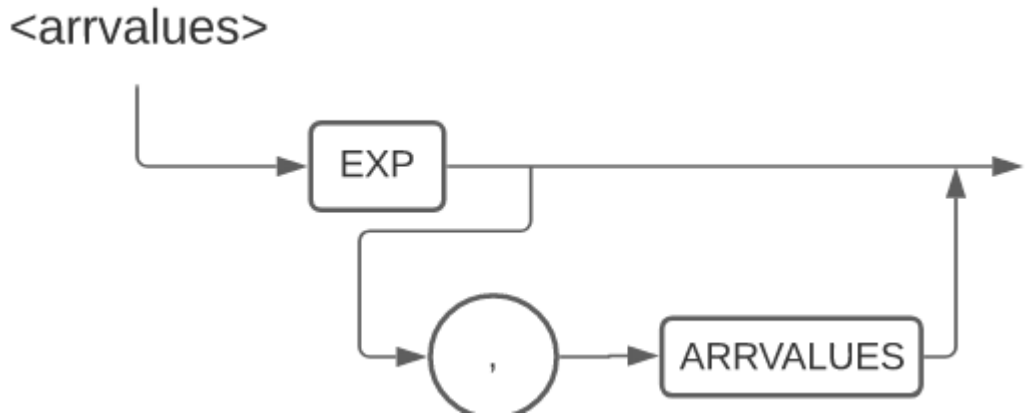
FuncEsp



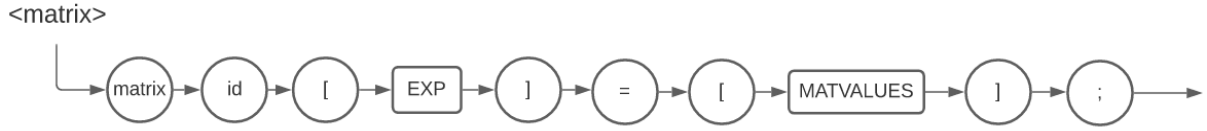
Array



Arrvalues

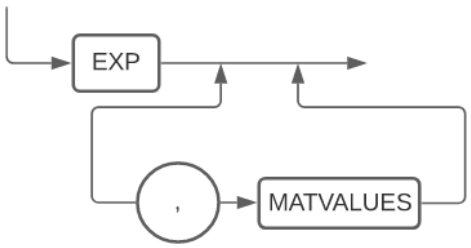


Matrix



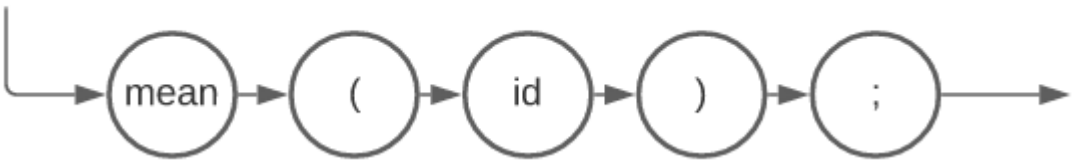
Matvalues

<matvalues>



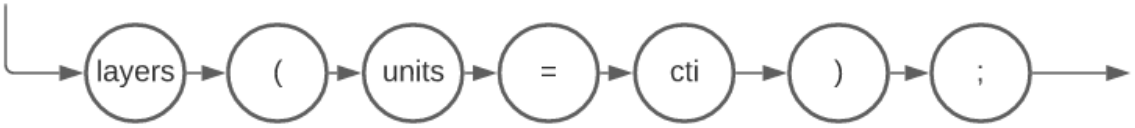
Mean

<mean>



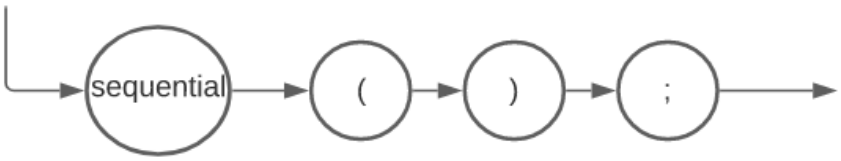
Layers

<layers>



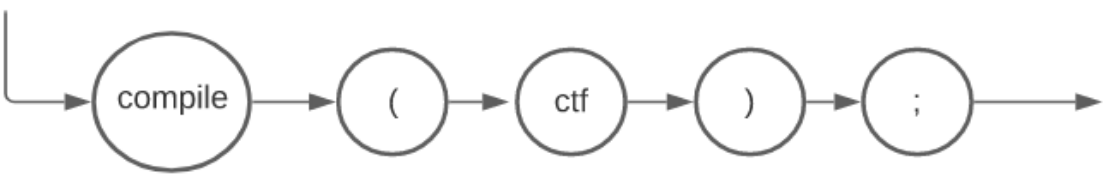
Sequential

<sequential>

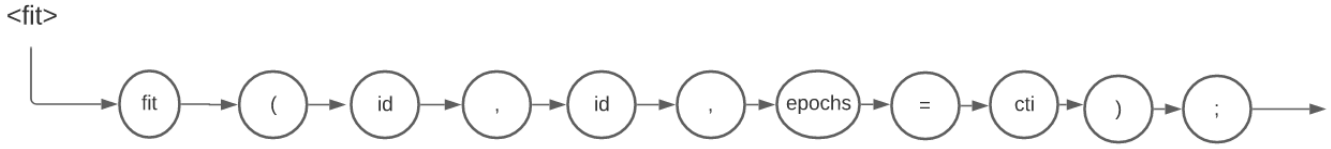


Compile

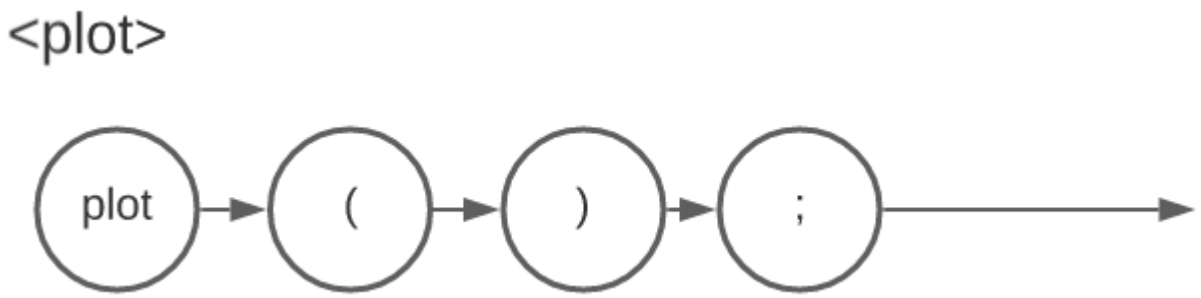
<compile>



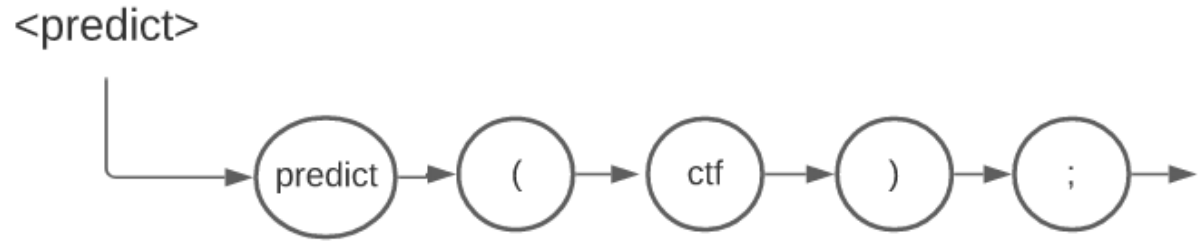
Fit



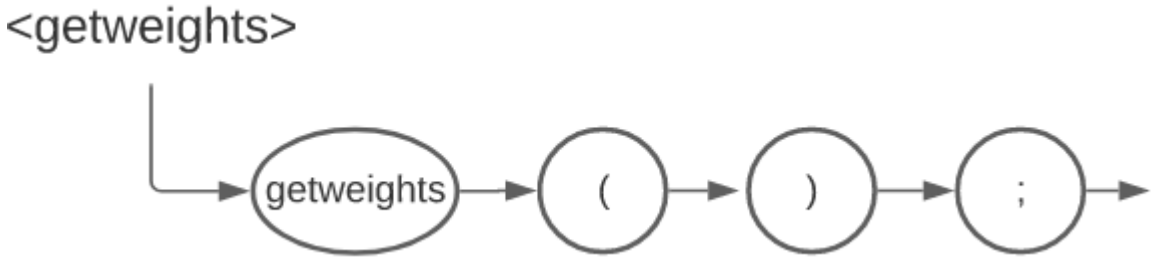
Plot



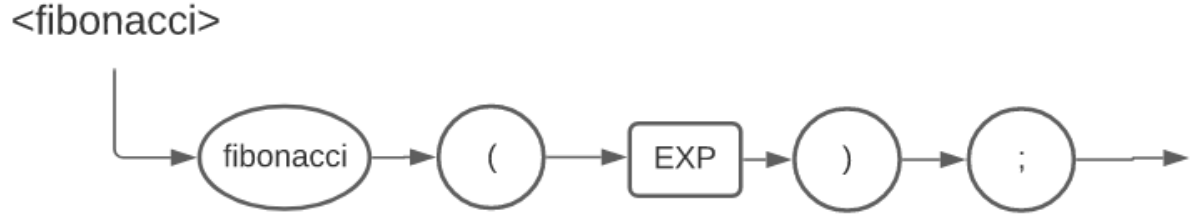
Predict



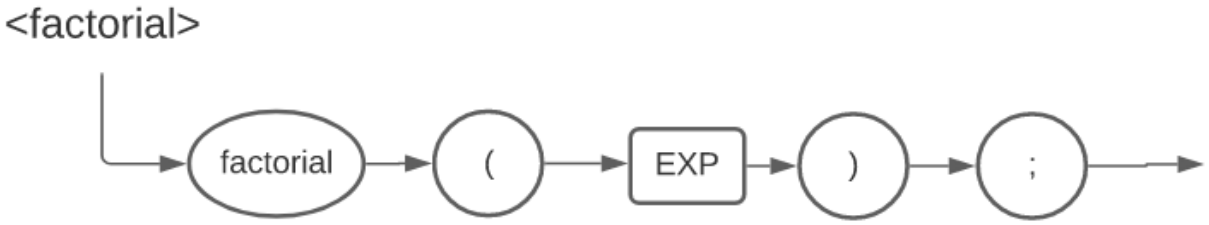
Getweights



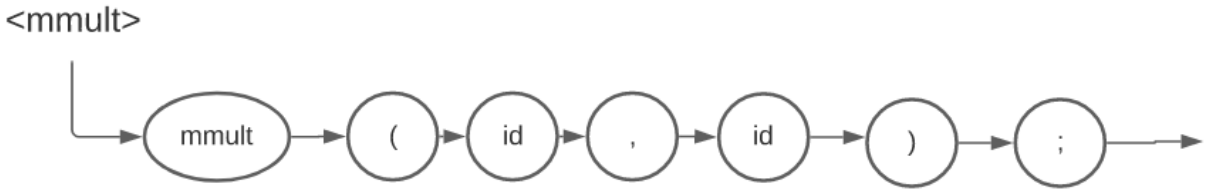
Fibonacci



Factorial



MatrixMult



Descripción de la Administración de Memoria en Compilación

La memoria se maneja mediante tablas, las cuales agregan las variables locales y globales dentro de sus respectivas tablas y les asigna su memoria correspondiente. De igual manera se hace una copia de la tabla actual y se integra un array. Al realizarse también se agrega el valor de las variables junto con su espacio de memoria a la tabla. Lo cual se ve de la siguiente manera:

```python
#Funcion que agrega variable local a tabla y asigna memoria
def add_var_local(name, type, currFunc):
    if(type=="int"):
        var_mem = local_var_table["counters"][type] + localInt
        local_var_table["variables"]["varInt"][name] = [var_mem]
    elif(type=="float"):
        var_mem = local_var_table["counters"][type] + localFloat
        local_var_table["variables"]["varFloat"][name] = [var_mem]
    #print("memoria:",var_mem)
    local_var_table["function"] = currFunc
    local_var_table["counters"][type] += 1


    sTypes.append(type)

#Funcion que agrega variable global a tabla y asigna memoria
def add_var_global(name, type):
    if(type=="int"):
        var_mem = var_table["global"]["counters"][type] + globalInt
    elif(type=="float"):
        var_mem = var_table["global"]["counters"][type] + globalFloat
    #print("memoria:",var_mem)
    var_table["global"]["counters"][type] += 1
    var_table["global"]["variables"][type][name] = [var_mem]
    sTypes.append(type)

#Funcion que hace una copia de la tabla actual y la mete a el array de tablas locales
def copy_var_local():
    local_var_table_copy = copy.deepcopy(local_var_table)
    global localArray
    localArray.append(local_var_table_copy)

#Funcion que agrega el valor de una variable y su espacio de memoria a la tabla
def add_local_var_value(variable_type,direccion_memoria, var_assign):
    if variable_type=="int":
        local_var_table["values"]["varInt"][direccion_memoria]=var_assign
    elif variable_type=="float":
        local_var_table["values"]["varFloat"][direccion_memoria]=var_assign
```

Directorio de Funciones

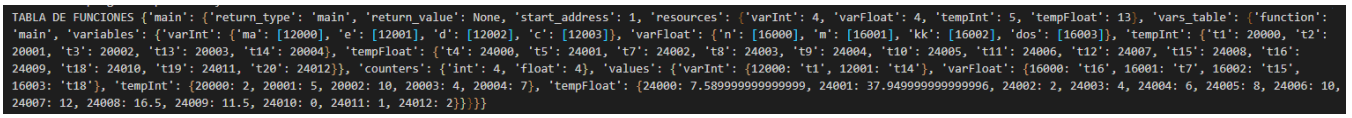
En la tabla de funciones se mapean todos los datos a procesar, mediante varios diccionarios lo cual permite acceder de una manera óptima a las tablas de variables. Mediante el scope local verifica cual es la función que está dentro, la cual almacena los parámetros y permite que sean accedidos en la tabla de funciones, cuadruplos, etc.

En la tabla de variables se almacenan los globales y locales y en cada uno sus respectivas variables, counters y values. Lo cual se desplegarán en el directorio de funciones al ejecutarse los cuádruplos.

```
functions_table = {}

def add_function(name, return_type, return_value, start_address, varInt, varFloat, tempInt, tempFloat,local_table):
    functions_table[name] = {
        "return_type": return_type,
        "return_value": return_value,
        "start_address": start_address,
        "resources": {
            "varInt": varInt,
            "varFloat": varFloat,
            "tempInt": tempInt,
            "tempFloat": tempFloat
        },
        "vars_table": local_table
    }
```

```
var_table = {
    "global": {
        "variables": {
            "int": {},
            "float": {}
        },
        "counters": {
            "int": 0,
            "float": 0
        },
        "values": {
            "int": {},
            "float": {}
        }
    }
}

local_var_table = {
    "function": "",
    "variables": {
        "varInt": {},
        "varFloat": {},
        "tempInt":{},
        "tempFloat":{}
    },
    "counters": {
        "int": 0,
        "float": 0
    },
    "values": {
        "varInt": {},
        "varFloat": {},
        "tempInt": {},
        "tempFloat": {}
    }
}
```

Ejemplo Directorio de funciones

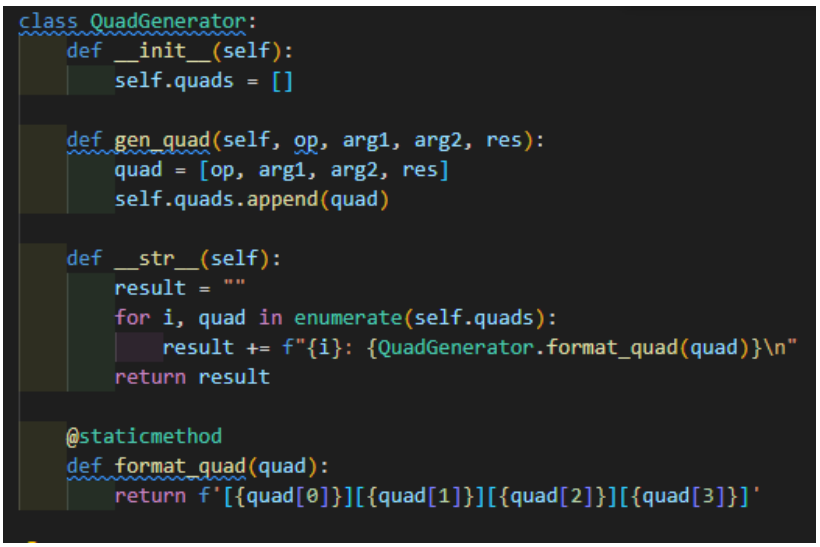



```
{
  "main": {
    "return_type": "main",
    "return_value": null,
    "start_address": 1,
    "resources": {
      "varInt": 4,
      "varFloat": 4,
      "tempInt": 5,
      "tempFloat": 13
    },
    "vars_table": {
      "function": "main",
      "variables": {
        "varInt": {
          "ma": [12000],
          "e": [12001],
          "d": [12002],
          "c": [12003]
        },
        "varFloat": {
          "n": [16000],
          "m": [16001],
          "kk": [16002],
          "dos": [16003]
        },
        "tempInt": {
          "t1": 20000,
          "t2": 20001,
          "t3": 20002,
          "t13": 20003,
          "t14": 20004
        },
        "tempFloat": {
          "t4": 24000,
          "t5": 24001,
          "t7": 24002,
          "t8": 24003,
          "t9": 24004,
          "t10": 24005,
          "t11": 24006,
          "t12": 24007,
          "t15": 24008,
```

```
          "t16": 24009,
          "t18": 24010,
          "t19": 24011,
          "t20": 24012
        }
      }
    },
    "counters": {
      "int": 4,
      "float": 4
    },
    "values": {
      "varInt": {
        "12000": "t1",
        "12001": "t14"
      },
      "varFloat": {
        "16000": "t16",
        "16001": "t7",
        "16002": "t15",
        "16003": "t18"
      },
      "tempInt": {
        "20000": 2,
        "20001": 5,
        "20002": 10,
        "20003": 4,
        "20004": 7
      },
      "tempFloat": {
        "24000": 7.59,
        "24001": 37.95,
        "24002": 2,
        "24003": 4,
        "24004": 6,
        "24005": 8,
        "24006": 10,
        "24007": 12,
        "24008": 16.5,
        "24009": 11.5,
        "24010": 0,
        "24011": 1,
        "24012": 2
      }
    }
  }
}
```

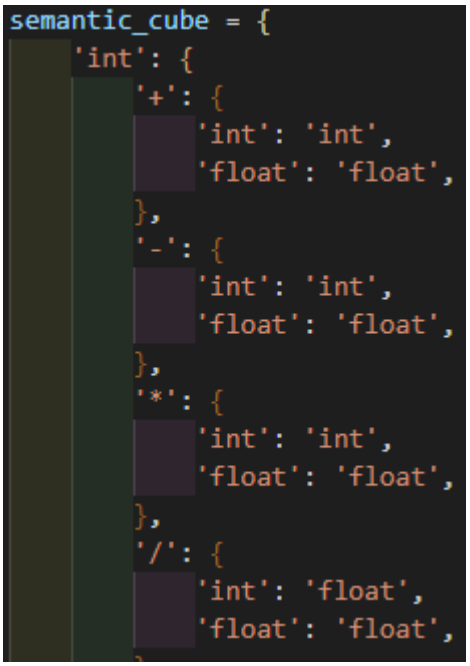
Pilas y Cubo Semántico

Cuando se manda a llamar la función que genera los cuádruplos, se almacenan en su array, lo cual después se integra a su respectiva pila.

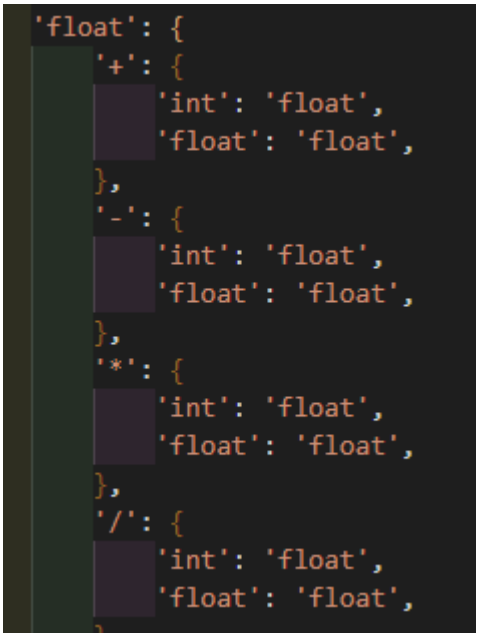


```python
class QuadGenerator:
    def __init__(self):
        self.quads = []

    def gen_quad(self, op, arg1, arg2, res):
        quad = [op, arg1, arg2, res]
        self.quads.append(quad)

    def __str__(self):
        result = ""
        for i, quad in enumerate(self.quads):
            result += f"{i}: {QuadGenerator.format_quad(quad)}\n"
        return result

    @staticmethod
    def format_quad(quad):
        return f'[{quad[0]}][{quad[1]}][{quad[2]}][{quad[3]}]'
```

El cubo semántico se maneja de manera en la que es un diccionario donde esta la combinación de los operadores operandos y el resultado, de la misma manera en la que están definida los cuádruplos. (op, arg1, arg2, res).



```python
semantic_cube = {
    'int': {
        '+': {
            'int': 'int',
            'float': 'float',
        },
        '-': {
            'int': 'int',
            'float': 'float',
        },
        '*': {
            'int': 'int',
            'float': 'float',
        },
        '/': {
            'int': 'float',
            'float': 'float',
```



```
'float': {
    '+': {
        'int': 'float',
        'float': 'float',
    },
    '-': {
        'int': 'float',
        'float': 'float',
    },
    '*': {
        'int': 'float',
        'float': 'float',
    },
    '/': {
        'int': 'float',
        'float': 'float',
```

Descripción de la Máquina Virtual

Objeto VM

```
#importa data de parser
functions_table=bixo.functions_table
sQuads=bixo.quadGen
gen_quad = bixo.quadGen.gen_quad
qCounter = bixo.qCounter
globalVarTable = bixo.var_table["global"]
```

Manejo de Memoria de Ejecución

Para el manejo de memoria de ejecución se utiliza una tabla de funciones la cual es el mapeo de todos los datos que se procesaron en el parser, lo cual incluye:
local_var_table: {
"function": "",
"variables": {
"varInt": {},
"varFloat": {},

```
        "tempInt":{},
        "tempFloat":{}
    },
    "counters": {
        "int": 0,
        "float": 0
    },
    "values": {
        "varInt": {},
        "varFloat": {},
        "tempInt": {},
        "tempFloat": {}
    }
}

functions_table: {
        "return_type": return_type,
        "return_value": return_value,
        "start_address": start_address,
        "resources": {
            "varInt": varInt,
            "varFloat": varFloat,
            "tempInt": tempInt,
            "tempFloat": tempFloat
        },
        "vars_table": local_table
    }
```

var_table que es una instancia de local_var_table que incluye todos los valores de local_var_table para cada función.

```
{
    "global": {
        "variables": {
            "int": {},
            "float": {}
        },
        "counters": {
            "int": 0,
            "float": 0
        },
        "values": {
            "int": {},
            "float": {}
        }
    }
}
```

Lo que crea una tabla de variables globales y se generan los cuádruplos referentes a la información almacenada.
La VM recibe todas las tablas de las funciones y las variables y las mapea utilizando las direcciones de memoria establecidas. Los datos recibidos por la VM son mapeados para optimizar espacio y recursos queda en un hashmap con la estructura de VARIABLE : [DIRECCIÓN DE MEMORIA, VALOR]. Para optimizar aún más recursos se realiza un segundo mapeo para asignar los valores de las temporales ya existentes a las variables que apuntan a dichos temporales y se asigna a otro hashmap llamado ValueMap.

Tabla de funciones:


```
{
   'que': {
      'return_type': 'void',
      'return_value': None,
      'start_address': 1,
      'resources': {
         'varInt': 1,
         'varFloat': 0,
         'tempInt': 0,
         'tempFloat': 0
      },
      'vars_table': {
         'function': 'que',
         'variables': {
            'varInt': {
               'c': [12000]
            },
            'varFloat': {},
            'tempInt': {},
            'tempFloat': {}
         },
         'counters': {
            'int': 1,
            'float': 0
         },
         'values': {
            'varInt': {},
            'varFloat': {},
            'tempInt': {},
            'tempFloat': {}
         }
      }
   },
   'main': {
      'return_type': 'main',
      'return_value': None,
      'start_address': 1,
      'resources': {
```

```
        'varInt': 5,
        'varFloat': 0,
        'tempInt': 18,
        'tempFloat': 0
    },
    'vars_table': {
        'function': 'main',
        'variables': {
            'varInt': {
                'ma': [12000],
                'e': [12001],
                'd': [12002],
                'c': [12003],
                'm': [12004]
            },
            'varFloat': {},
            'tempInt': {
                't1': 20000,
                't2': 20001,
                't3': 20002,
                't4': 20003,
                't5': 20004,
                't6': 20005,
                't7': 20006,
                't8': 20007,
                't9': 20008,
                't10': 20009,
                't11': 20010,
                't12': 20011,
                't13': 20012,
                't14': 20013,
                't15': 20014,
                't16': 20015,
                't17': 20016,
                't18': 20017
            },
            'tempFloat': {}
        },
        'counters': {
            'int': 5,
            'float': 0
        },
        'values': {
            'varInt': {
                12000: 't1',
                12003: 't18',
                12004: 't12',
                12001: 't15'
```

```
            },
            'varFloat': {},
            'tempInt': {
                20000: 2,
                20001: 8,
                20002: 9,
                20003: 14,
                20004: 6,
                20005: 1,
                20006: 0,
                20007: 0,
                20008: 1,
                20009: 1,
                20010: 25,
                20011: 29,
                20012: 2,
                20013: 8,
                20014: 10,
                20015: 6,
                20016: 11,
                20017: 17
            },
            'tempFloat': {}
        }
    }
}
```

Tabla de variables globales:

```
{
    'global': {
        'variables': {
            'int': {},
            'float': {}
        },
        'counters': {
            'int': 0,
            'float': 0
        },
        'values': {
            'int': {},
            'float': {}
        }
    }
}
```

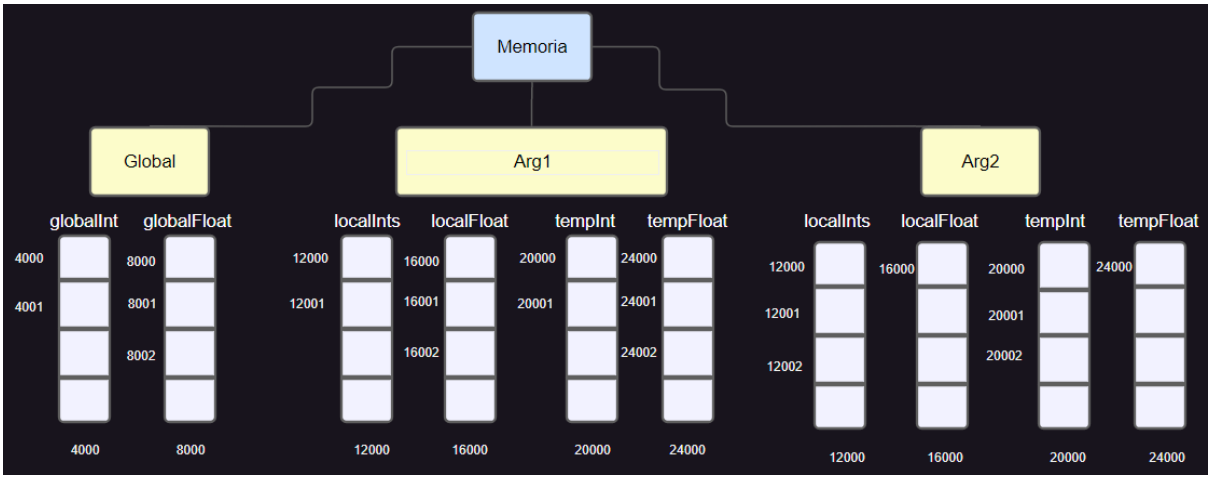
HashMap

```
{
    'global': {},
    'que': {
        'c': (12000, '')
    },
    'main': {
        'ma': (12000, 't1'),
        'e': (12001, 't15'),
        'd': (12002, ''),
        'c': (12003, 't18'),
        'm': (12004, 't12'),
        't1': (20000, 2),
        't2': (20001, 8),
        't3': (20002, 9),
        't4': (20003, 14),
        't5': (20004, 6),
        't6': (20005, 1),
        't7': (20006, 0),
        't8': (20007, 0),
        't9': (20008, 1),
        't10': (20009, 1),
        't11': (20010, 25),
        't12': (20011, 29),
        't13': (20012, 2),
        't14': (20013, 8),
        't15': (20014, 10),
        't16': (20015, 6),
        't17': (20016, 11),
        't18': (20017, 17)
    }
}
```

ValueMap

```
{
    'global': {},
    'que': {
        'c': (12000, '')
    },
    'main': {
        'ma': (12000, 2),
        'e': (12001, 10),
        'd': (12002, ''),
        'c': (12003, 17),
        'm': (12004, 29),
        't1': (20000, 2),
        't2': (20001, 8),
        't3': (20002, 9),
        't4': (20003, 14),
```

```
        't5': (20004, 6),
        't6': (20005, 1),
        't7': (20006, 0),
        't8': (20007, 0),
        't9': (20008, 1),
        't10': (20009, 1),
        't11': (20010, 25),
        't12': (20011, 29),
        't13': (20012, 2),
        't14': (20013, 8),
        't15': (20014, 10),
        't16': (20015, 6),
        't17': (20016, 11),
        't18': (20017, 17)
    }
}
```



Pruebas del funcionamiento del lenguaje

Prueba.bixo

Código

```
program id;
var float col, que;
var int pepe, rodrigo, juan;

function void cerrar (float c, float y){
    var int l;
    var float cm;
```

```
    l = 3;
    pepe=3;
    cm = 5+4+5*4*2/2;
    var float cmmi;
    cmmi=cm+4*2.2;
    print(cmmi);
    read(l);
    print(l);
    print(c);
    print(y);
}
main(){
    var int ma;
    var float klo;
    klo=5*3.3;
    cerrar(klo, 8.8)
    ma=1+1;
    print(ma);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][19]
1: [FUNCSTART][cerrar][None][1]
2: [=][3][None][l]
3: [=][3][None][pepe]
4: [+][5][4][t1]
5: [*][5][4][t2]
6: [*][t2][2][t3]
7: [/][t3][2][t4]
8: [+][t1][t4][t5]
9: [=][t5][None][cm]
10: [*][4][2.2][t6]
11: [+][cm][t6][t7]
12: [=][t7][None][cmmi]
13: [print][None][None][cmmi]
14: [read][None][None][l]
15: [print][None][None][l]
16: [print][None][None][c]
17: [print][None][None][y]
18: [FUNCEND][cerrar][None][None]
19: [FUNCSTART][main][None][19]
20: [*][5][3.3][t1]
21: [=][t1][None][klo]
22: [ERA][None][None][cerrar]
23: [PARAM][klo][None][par1]
24: [PARAM][8.8][None][par2]
25: [GOSUB][None][None][1]
26: [+][1][1][t2]
27: [=][t2][None][ma]
28: [print][None][None][ma]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 19)
Quad 19 ['FUNCSTART', 'main', None, 19]
Quad 20 ['*', 5, 3.3, 't1']
Quad 21 ['=', 't1', None, 'klo']
Quad 22 ['ERA', None, None, 'cerrar']
Quad 23 ['PARAM', 'klo', None, 'par1']
Quad 24 ['PARAM', 8.8, None, 'par2']
Quad 25 ['GOSUB', None, None, 1]
Quad 1 ['FUNCSTART', 'cerrar', None, 1]
Quad 2 ['=', 3, None, 'l']
Quad 3 ['=', 3, None, 'pepe']
Quad 4 ['+', 5, 4, 't1']
Quad 5 ['*', 5, 4, 't2']
Quad 6 ['*', 't2', 2, 't3']
Quad 7 ['/', 't3', 2, 't4']
Quad 8 ['+', 't1', 't4', 't5']
Quad 9 ['=', 't5', None, 'cm']
Quad 10 ['*', 4, 2.2, 't6']
Quad 11 ['+', 'cm', 't6', 't7']
Quad 12 ['=', 't7', None, 'cmmi']
Quad 13 ['print', None, None, 'cmmi']
37.8
Quad 14 ['read', None, None, 'l']
9
Quad 15 ['print', None, None, 'l']
9
Quad 16 ['print', None, None, 'c']
16.5
Quad 17 ['print', None, None, 'y']
8.8
Quad 18 ['FUNCEND', 'cerrar', None, None]
Quad 26 ['+', 1, 1, 't2']
Quad 27 ['=', 't2', None, 'ma']
Quad 28 ['print', None, None, 'ma']
2
```

Resultados

37.8
9

16.5
8.8
2

Prueba2.bixo

Código

```
program id;
var int i;
function int cerrar(){
    var float c;
    var int d;
    var int kl;
    c=3/2+4*4*6.3+6+7.7/34.4;
    kl=3+5;
    return kl;
}
function float abrir (float k, float c){
    var float a;
    var float b;
    a=3+4*5+6.6+9-4.4;
    b=2.2+3+4*5;
    print(a,b);
    return b;
}
main(){
    var float ce;
    ce=abrir(3.4,2.2);
    var int ma;
    var float kle;
    kle=3+4+5.5;
    print(ce);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][27]
1: [FUNCSTART][cerrar][None][1]
2: [/][3][2][t1]
3: [*][4][4][t2]
4: [*][t2][6.3][t3]
5: [+][t1][t3][t4]
6: [+][t4][6][t5]
7: [/][7.7][34.4][t6]
8: [+][t5][t6][t7]
9: [=][t7][None][c]
10: [+][3][5][t8]
11: [=][t8][None][kl]
12: [FUNCEND][cerrar][None][kl]
13: [FUNCSTART][abrir][None][13]
14: [*][4][5][t1]
15: [+][3][t1][t2]
16: [+][t2][6.6][t3]
17: [+][t3][9][t4]
18: [-][t4][4.4][t5]
19: [=][t5][None][a]
20: [+][2.2][3][t6]
21: [*][4][5][t7]
22: [+][t6][t7][t8]
23: [=][t8][None][b]
24: [print][None][None][a]
25: [print][None][None][b]
26: [FUNCEND][abrir][None][b]
27: [FUNCSTART][main][None][27]
28: [ERA][None][None][abrir]
29: [PARAM][3.4][None][par1]
30: [PARAM][2.2][None][par2]
31: [GOSUB][None][None][13]
32: [ASSIGNFUNC][abrir][main][t1]
33: [=][t1][None][ce]
34: [+][3][4][t2]
35: [+][t2][5.5][t3]
36: [=][t3][None][kle]
37: [print][None][None][ce]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 27)
Quad 27 ['FUNCSTART', 'main', None, 27]
Quad 28 ['ERA', None, None, 'abrir']
Quad 29 ['PARAM', 3.4, None, 'par1']
Quad 30 ['PARAM', 2.2, None, 'par2']
Quad 31 ['GOSUB', None, None, 13]
Quad 13 ['FUNCSTART', 'abrir', None, 13]
Quad 14 ['*', 4, 5, 't1']
Quad 15 ['+', 3, 't1', 't2']
Quad 16 ['+', 't2', 6.6, 't3']
Quad 17 ['+', 't3', 9, 't4']
Quad 18 ['-', 't4', 4.4, 't5']
Quad 19 ['=', 't5', None, 'a']
Quad 20 ['+', 2.2, 3, 't6']
Quad 21 ['*', 4, 5, 't7']
Quad 22 ['+', 't6', 't7', 't8']
Quad 23 ['=', 't8', None, 'b']
Quad 24 ['print', None, None, 'a']
34.2
Quad 25 ['print', None, None, 'b']
25.2
Quad 26 ['FUNCEND', 'abrir', None, 'b']
Quad 32 ['ASSIGNFUNC', 'abrir', 'main', 't1']
Quad 33 ['=', 't1', None, 'ce']
Quad 34 ['+', 3, 4, 't2']
Quad 35 ['+', 't2', 5.5, 't3']
Quad 36 ['=', 't3', None, 'kle']
Quad 37 ['print', None, None, 'ce']
25.2
```

Resultados

34.2
25.2
25.2

Prueba3.bixo

Código

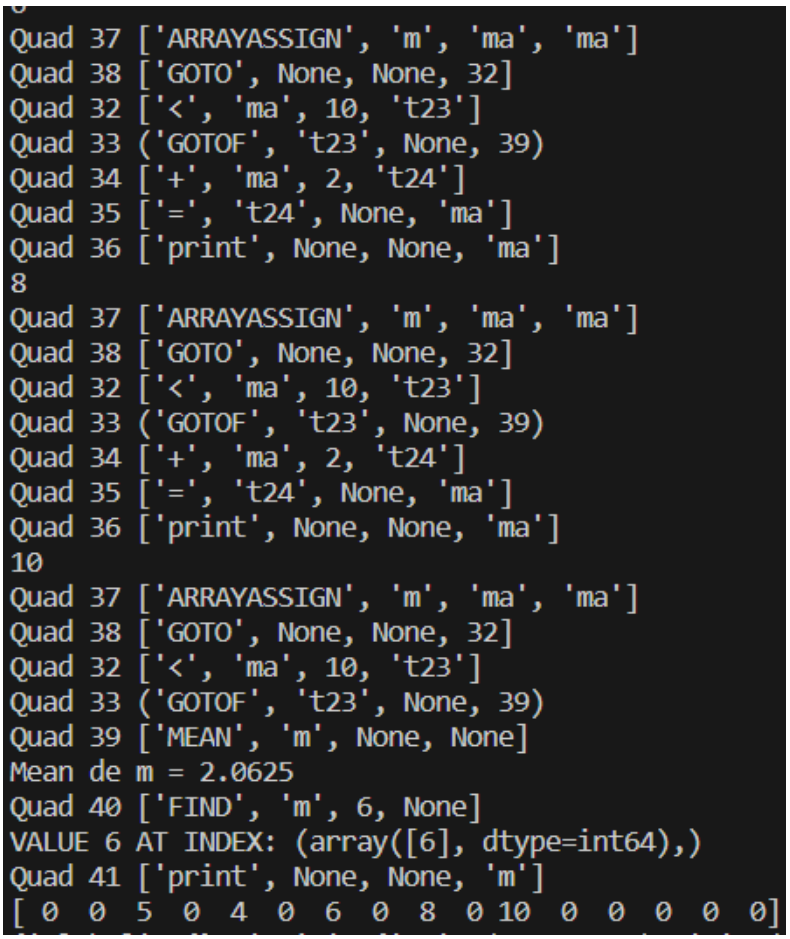
```
program pruebarray;

main(){
    var int ma;
    var int c,d,e;
    var float n;
    ma=1+1;
    e=2+3+5;
    n=2.3*3.3*5;
    array m[15]=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
    m[2]=5;
    print(m);
    print(m[2]);
    while (ma<10){
    ma=ma+2;
    print(ma);
    m[ma]=ma;
    };
    mean(m);
    find(m,6);
    print(m);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][1]
1: [FUNCSTART][main][None][1]
2: [+][1][1][t1]
3: [=][t1][None][ma]
4: [+][2][3][t2]
5: [+][t2][5][t3]
6: [=][t3][None][e]
7: [*][2.3][3.3][t4]
8: [*][t4][5][t5]
9: [=][t5][None][n]
10: [ARRAY][m][15][None]
11: [ARRAYSTART][t7][16001][m]
12: [ARRAYFILL][m[0]][t7][m]
13: [ARRAYFILL][m[1]][t8][m]
14: [ARRAYFILL][m[2]][t9][m]
15: [ARRAYFILL][m[3]][t10][m]
16: [ARRAYFILL][m[4]][t11][m]
17: [ARRAYFILL][m[5]][t12][m]
18: [ARRAYFILL][m[6]][t13][m]
19: [ARRAYFILL][m[7]][t14][m]
20: [ARRAYFILL][m[8]][t15][m]
21: [ARRAYFILL][m[9]][t16][m]
22: [ARRAYFILL][m[10]][t17][m]
23: [ARRAYFILL][m[11]][t18][m]
24: [ARRAYFILL][m[12]][t19][m]
25: [ARRAYFILL][m[13]][t20][m]
26: [ARRAYFILL][m[14]][t21][m]
27: [ARRAYFILL][m[15]][t22][m]
28: [ARRAYEND][t22][16001][m]
29: [ARRAYASSIGN][m][2][5]
30: [print][None][None][m]
31: [ARRAYPRINT][None][None][m[2]]
32: [<][ma][10][t23]
33: [GOTOF][t23][None][39]
34: [+][ma][2][t24]
35: [=][t24][None][ma]
36: [print][None][None][ma]
37: [ARRAYASSIGN][m][ma][ma]
38: [GOTO][None][None][32]
39: [MEAN][m][None][None]
40: [FIND][m][6][None]
41: [print][None][None][m]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 1)
Quad 1 ['FUNCSTART', 'main', None, 1]
Quad 2 ['+', 1, 1, 't1']
Quad 3 ['=', 't1', None, 'ma']
Quad 4 ['+', 2, 3, 't2']
Quad 5 ['+', 't2', 5, 't3']
Quad 6 ['=', 't3', None, 'e']
Quad 7 ['*', 2.3, 3.3, 't4']
Quad 8 ['*', 't4', 5, 't5']
Quad 9 ['=', 't5', None, 'n']
Quad 10 ['ARRAY', 'm', 15, None]
Quad 11 ['ARRAYSTART', 't7', 16001, 'm']
Quad 12 ['ARRAYFILL', 'm[0]', 't7', 'm']
Quad 13 ['ARRAYFILL', 'm[1]', 't8', 'm']
Quad 14 ['ARRAYFILL', 'm[2]', 't9', 'm']
Quad 15 ['ARRAYFILL', 'm[3]', 't10', 'm']
Quad 16 ['ARRAYFILL', 'm[4]', 't11', 'm']
Quad 17 ['ARRAYFILL', 'm[5]', 't12', 'm']
Quad 18 ['ARRAYFILL', 'm[6]', 't13', 'm']
Quad 19 ['ARRAYFILL', 'm[7]', 't14', 'm']
Quad 20 ['ARRAYFILL', 'm[8]', 't15', 'm']
Quad 21 ['ARRAYFILL', 'm[9]', 't16', 'm']
Quad 22 ['ARRAYFILL', 'm[10]', 't17', 'm']
Quad 23 ['ARRAYFILL', 'm[11]', 't18', 'm']
Quad 24 ['ARRAYFILL', 'm[12]', 't19', 'm']
Quad 25 ['ARRAYFILL', 'm[13]', 't20', 'm']
Quad 26 ['ARRAYFILL', 'm[14]', 't21', 'm']
Quad 27 ['ARRAYFILL', 'm[15]', 't22', 'm']
Quad 28 ['ARRAYEND', 't22', 16001, 'm']
Quad 29 ['ARRAYASSIGN', 'm', 2, 5]
Quad 30 ['print', None, None, 'm']
[0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0]
Quad 31 ['ARRAYPRINT', None, None, 'm[2]']
5
Quad 32 ['<', 'ma', 10, 't23']
Quad 33 ('GOTOF', 't23', None, 39)
Quad 34 ['+', 'ma', 2, 't24']
Quad 32 ['<', 'ma', 10, 't23']
Quad 33 ('GOTOF', 't23', None, 39)
Quad 34 ['+', 'ma', 2, 't24']
Quad 35 ['=', 't24', None, 'ma']
Quad 36 ['print', None, None, 'ma']
6
```



Resultados

[0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0]
5
6
8
10
[0 0 5 0 4 0 6 0 8 0 10 0 0 0 0 0]

Prueba4.bixo

Código

```
program pruebamatrix;
main(){
    matrix m[3][3]=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16];
    var float kk;
    kk=3.3*5;
    matrix z[3][3]=[2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32];
    z[2][1]=5;
    mmult(m,z);
```

```
    print(z[2][1]);
    print(z);
    mean(m);
    sort(m);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][1]
1: [FUNCSTART][main][None][1]
2: [MATRIX][m][3][3]
3: [MATRIXSTART][t2][16000][m]
4: [MATRIXFILL][m[0][0]][t2][m]
5: [MATRIXFILL][m[0][1]][t3][m]
6: [MATRIXFILL][m[0][2]][t4][m]
7: [MATRIXFILL][m[0][3]][t5][m]
8: [MATRIXFILL][m[1][0]][t6][m]
9: [MATRIXFILL][m[1][1]][t7][m]
10: [MATRIXFILL][m[1][2]][t8][m]
11: [MATRIXFILL][m[1][3]][t9][m]
12: [MATRIXFILL][m[2][0]][t10][m]
13: [MATRIXFILL][m[2][1]][t11][m]
14: [MATRIXFILL][m[2][2]][t12][m]
15: [MATRIXFILL][m[2][3]][t13][m]
16: [MATRIXFILL][m[3][0]][t14][m]
17: [MATRIXFILL][m[3][1]][t15][m]
18: [MATRIXFILL][m[3][2]][t16][m]
19: [MATRIXFILL][m[3][3]][t17][m]
20: [MATRIXEND][t17][16000][m]
21: [*][3.3][5][t18]
22: [=][t18][None][kk]
23: [MATRIX][z][3][3]
24: [MATRIXSTART][t20][16002][z]
25: [MATRIXFILL][z[0][0]][t20][z]
26: [MATRIXFILL][z[0][1]][t21][z]
27: [MATRIXFILL][z[0][2]][t22][z]
28: [MATRIXFILL][z[0][3]][t23][z]
29: [MATRIXFILL][z[1][0]][t24][z]
30: [MATRIXFILL][z[1][1]][t25][z]
31: [MATRIXFILL][z[1][2]][t26][z]
32: [MATRIXFILL][z[1][3]][t27][z]
33: [MATRIXFILL][z[2][0]][t28][z]
34: [MATRIXFILL][z[2][1]][t29][z]
35: [MATRIXFILL][z[2][2]][t30][z]
36: [MATRIXFILL][z[2][3]][t31][z]
37: [MATRIXFILL][z[3][0]][t32][z]
38: [MATRIXFILL][z[3][1]][t33][z]
39: [MATRIXFILL][z[3][2]][t34][z]
40: [MATRIXFILL][z[3][3]][t35][z]
41: [MATRIXEND][t35][16002][z]
42: [MATRIXASSIGN][z][[2, 1]][5]
43: [MATRIXMULT][m][z][None]
```

```
44: [MATRIXPRINT][None][None][z[2][1]]
45: [print][None][None][z]
46: [MEAN][m][None][None]
47: [SORT][m][None][None]
```

Orden de Ejecución de Cuádruplos

```
Quad 1 ['FUNCSTART', 'main', None, 1]
Quad 2 ['MATRIX', 'm', 3, 3]
Quad 3 ['MATRIXSTART', 't2', 16000, 'm']
Quad 4 ['MATRIXFILL', 'm[0][0]', 't2', 'm']
Quad 5 ['MATRIXFILL', 'm[0][1]', 't3', 'm']
Quad 6 ['MATRIXFILL', 'm[0][2]', 't4', 'm']
Quad 7 ['MATRIXFILL', 'm[0][3]', 't5', 'm']
Quad 8 ['MATRIXFILL', 'm[1][0]', 't6', 'm']
Quad 9 ['MATRIXFILL', 'm[1][1]', 't7', 'm']
Quad 10 ['MATRIXFILL', 'm[1][2]', 't8', 'm']
Quad 11 ['MATRIXFILL', 'm[1][3]', 't9', 'm']
Quad 12 ['MATRIXFILL', 'm[2][0]', 't10', 'm']
Quad 13 ['MATRIXFILL', 'm[2][1]', 't11', 'm']
Quad 14 ['MATRIXFILL', 'm[2][2]', 't12', 'm']
Quad 15 ['MATRIXFILL', 'm[2][3]', 't13', 'm']
Quad 16 ['MATRIXFILL', 'm[3][0]', 't14', 'm']
Quad 17 ['MATRIXFILL', 'm[3][1]', 't15', 'm']
Quad 18 ['MATRIXFILL', 'm[3][2]', 't16', 'm']
Quad 19 ['MATRIXFILL', 'm[3][3]', 't17', 'm']
Quad 20 ['MATRIXEND', 't17', 16000, 'm']
```

```
Quad 21 ['*', 3.3, 5, 't18']
Quad 22 ['=', 't18', None, 'kk']
Quad 23 ['MATRIX', 'z', 3, 3]
Quad 24 ['MATRIXSTART', 't20', 16002, 'z']
Quad 25 ['MATRIXFILL', 'z[0][0]', 't20', 'z']
Quad 26 ['MATRIXFILL', 'z[0][1]', 't21', 'z']
Quad 27 ['MATRIXFILL', 'z[0][2]', 't22', 'z']
Quad 28 ['MATRIXFILL', 'z[0][3]', 't23', 'z']
Quad 29 ['MATRIXFILL', 'z[1][0]', 't24', 'z']
Quad 30 ['MATRIXFILL', 'z[1][1]', 't25', 'z']
Quad 31 ['MATRIXFILL', 'z[1][2]', 't26', 'z']
Quad 32 ['MATRIXFILL', 'z[1][3]', 't27', 'z']
Quad 33 ['MATRIXFILL', 'z[2][0]', 't28', 'z']
Quad 34 ['MATRIXFILL', 'z[2][1]', 't29', 'z']
Quad 35 ['MATRIXFILL', 'z[2][2]', 't30', 'z']
Quad 36 ['MATRIXFILL', 'z[2][3]', 't31', 'z']
Quad 37 ['MATRIXFILL', 'z[3][0]', 't32', 'z']
Quad 38 ['MATRIXFILL', 'z[3][1]', 't33', 'z']
Quad 39 ['MATRIXFILL', 'z[3][2]', 't34', 'z']
Quad 40 ['MATRIXFILL', 'z[3][3]', 't35', 'z']
Quad 41 ['MATRIXEND', 't35', 16002, 'z']
```

```
Quad 43 ['MATRIXMULT', 'm', 'z', None]
MULTIPLICACION DE m * z ES = [[ 180  155  220  240]
 [ 404  351  508  560]
 [ 628  547  796  880]
 [ 852  743 1084 1200]]
Quad 44 ['MATRIXPRINT', None, None, 'z[2][1]']
5
Quad 45 ['print', None, None, 'z']
[[ 2  4  6  8]
 [10 12 14 16]
 [18  5 22 24]
 [26 28 30 32]]
Quad 46 ['MEAN', 'm', None, None]
Mean de m = 8.5
Quad 47 ['SORT', 'm', None, None]
Sorted matrix m = [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Resultados

```
MULTIPLICACION DE m * z ES =
[[ 180  155  220  240]
 [ 404  351  508  560]
 [ 628  547  796  880]
 [ 852  743 1084 1200]]
5
[[ 2  4  6  8]
 [10 12 14 16]
 [18  5 22 24]
 [26 28 30 32]]
Mean de m = 8.5
Sorted matrix m =
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Pruebaif1.bixo

Código

```
program pruebaif;
var int i;
function float funcuno(){
    var int a;
    var float b;
    a=3+4;
    b=2.2+3+4.4;
    return b;
}
main(){
    var int ma;
    var float k,j;
    ma=2+5+6*5;
    j=funcuno();
    if(3==4){
    print(j);
    }else {
    print(ma);
    };
    k=ma;
}
```

Cuádruplos

```
1: [FUNCSTART][funcuno][None][1]
2: [+][3][4][t1]
3: [=][t1][None][a]
4: [+][2.2][3][t2]
5: [+][t2][4.4][t3]
6: [=][t3][None][b]
7: [FUNCEND][funcuno][None][b]
8: [FUNCSTART][main][None][8]
9: [+][2][5][t1]
10: [*][6][5][t2]
11: [+][t1][t2][t3]
12: [=][t3][None][ma]
13: [ERA][None][None][funcuno]
14: [GOSUB][None][None][1]
15: [ASSIGNFUNC][funcuno][main][t4]
16: [=][t4][None][j]
17: [==][3][4][t5]
18: [GOTOF][t5][None][21]
19: [print][None][None][j]
20: [GOTO][None][None][22]
21: [print][None][None][ma]
22: [=][ma][None][k]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 8)
Quad 8 ['FUNCSTART', 'main', None, 8]
Quad 9 ['+', 2, 5, 't1']
Quad 10 ['*', 6, 5, 't2']
Quad 11 ['+', 't1', 't2', 't3']
Quad 12 ['=', 't3', None, 'ma']
Quad 13 ['ERA', None, None, 'funcuno']
Quad 14 ['GOSUB', None, None, 1]
Quad 1 ['FUNCSTART', 'funcuno', None, 1]
Quad 2 ['+', 3, 4, 't1']
Quad 3 ['=', 't1', None, 'a']
Quad 4 ['+', 2.2, 3, 't2']
Quad 5 ['+', 't2', 4.4, 't3']
Quad 6 ['=', 't3', None, 'b']
Quad 7 ['FUNCEND', 'funcuno', None, 'b']
Quad 15 ['ASSIGNFUNC', 'funcuno', 'main', 't4']
Quad 16 ['=', 't4', None, 'j']
Quad 17 ['==', 3, 4, 't5']
Quad 18 ('GOTOF', 't5', None, 21)
Quad 21 ['print', None, None, 'ma']
37.0
Quad 22 ['=', 'ma', None, 'k']
```

Resultados

37

Pruebaif2.bixo

Código

```
program pruebaif;
var int i;
function int que(int a, int b){
    var int m;
    m=10+5;
    return m;
}
main(){
    var int c,t,y;
    c=5+3+1+5;
    y=5-1;
    if(c!=y & y==4){
    y=50;
    }else {
    y=100;
    };
    print(y);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][5]
1: [FUNCSTART][que][None][1]
2: [+][10][5][t1]
3: [=][t1][None][m]
4: [FUNCEND][que][None][m]
5: [FUNCSTART][main][None][5]
6: [+][5][3][t1]
7: [+][t1][1][t2]
8: [+][t2][5][t3]
9: [=][t3][None][c]
10: [-][5][1][t4]
11: [=][t4][None][y]
12: [!=][c][y][t5]
13: [==][y][4][t6]
14: [AND][t5][t6][t7]
15: [GOTOF][t7][None][18]
16: [=][50][None][y]
17: [GOTO][None][None][19]
18: [=][100][None][y]
19: [print][None][None][y]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 5)
Quad 5 ['FUNCSTART', 'main', None, 5]
Quad 6 ['+', 5, 3, 't1']
Quad 7 ['+', 't1', 1, 't2']
Quad 8 ['+', 't2', 5, 't3']
Quad 9 ['=', 't3', None, 'c']
Quad 10 ['-', 5, 1, 't4']
Quad 11 ['=', 't4', None, 'y']
Quad 12 ['!=', 'c', 'y', 't5']
Quad 13 ['==', 'y', 4, 't6']
Quad 14 ['AND', 't5', 't6', 't7']
Quad 15 ('GOTOF', 't7', None, 18)
Quad 16 ['=', 50, None, 'y']
Quad 17 ('GOTO', None, None, 19)
Quad 19 ['print', None, None, 'y']
50
```

Resultados

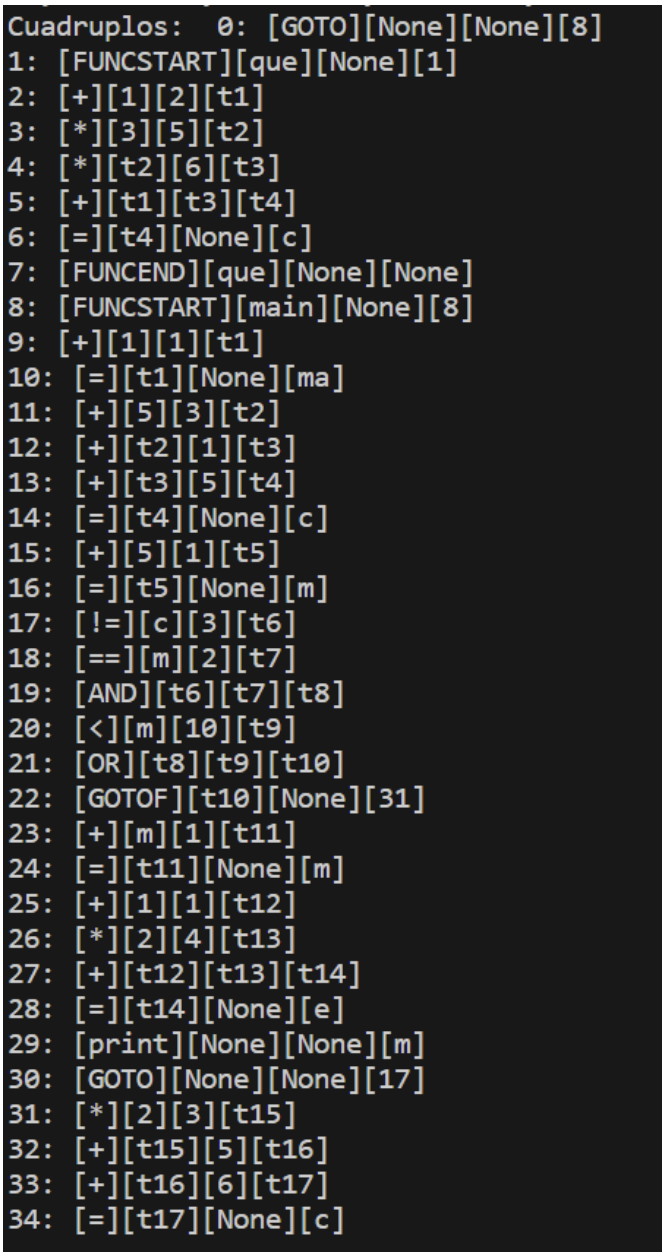
50

Pruebawhile.bixo

Código

```
program pruebawhile;
function void que(){
    var int c;
    c=1+2+3*5*6;
}
main(){
    var int ma;
    ma=1+1;
    var int c,d,e;
    var int m;
    c=5+3+1+5;
    m=5+1;
    while(c!=3 & m==2 | m<10){
    m=m+1;
    e=1+1+2*4;
    print(m);
    };
    c=2*3+5+6;
}
```

Cuádruplos


Cuadruplos: 0: [GOTO][None][None][8]
1: [FUNCSTART][que][None][1]
2: [+][1][2][t1]
3: [*][3][5][t2]
4: [*][t2][6][t3]
5: [+][t1][t3][t4]
6: [=][t4][None][c]
7: [FUNCEND][que][None][None]
8: [FUNCSTART][main][None][8]
9: [+][1][1][t1]
10: [=][t1][None][ma]
11: [+][5][3][t2]
12: [+][t2][1][t3]
13: [+][t3][5][t4]
14: [=][t4][None][c]
15: [+][5][1][t5]
16: [=][t5][None][m]
17: [!=][c][3][t6]
18: [==][m][2][t7]
19: [AND][t6][t7][t8]
20: [<][m][10][t9]
21: [OR][t8][t9][t10]
22: [GOTOF][t10][None][31]
23: [+][m][1][t11]
24: [=][t11][None][m]
25: [+][1][1][t12]
26: [*][2][4][t13]
27: [+][t12][t13][t14]
28: [=][t14][None][e]
29: [print][None][None][m]
30: [GOTO][None][None][17]
31: [*][2][3][t15]
32: [+][t15][5][t16]
33: [+][t16][6][t17]
34: [=][t17][None][c]

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 8)
Quad 8 ['FUNCSTART', 'main', None, 8]
Quad 9 ['+', 1, 1, 't1']
Quad 10 ['=', 't1', None, 'ma']
Quad 11 ['+', 5, 3, 't2']
Quad 12 ['+', 't2', 1, 't3']
Quad 13 ['+', 't3', 5, 't4']
Quad 14 ['=', 't4', None, 'c']
Quad 15 ['+', 5, 1, 't5']
Quad 16 ['=', 't5', None, 'm']
Quad 17 ['!=', 'c', 3, 't6']
Quad 18 ['==', 'm', 2, 't7']
Quad 19 ['AND', 't6', 't7', 't8']
Quad 20 ['<', 'm', 10, 't9']
Quad 21 ['OR', 't8', 't9', 't10']
Quad 22 ('GOTOF', 't10', None, 31)
Quad 23 ['+', 'm', 1, 't11']
Quad 24 ['=', 't11', None, 'm']
Quad 25 ['+', 1, 1, 't12']
Quad 26 ['*', 2, 4, 't13']
Quad 27 ['+', 't12', 't13', 't14']
Quad 28 ['=', 't14', None, 'e']
Quad 29 ['print', None, None, 'm']
7
Quad 30 ['GOTO', None, None, 17]
Quad 17 ['!=', 'c', 3, 't6']
Quad 18 ['==', 'm', 2, 't7']
Quad 19 ['AND', 't6', 't7', 't8']
Quad 20 ['<', 'm', 10, 't9']
Quad 21 ['OR', 't8', 't9', 't10']
Quad 22 ('GOTOF', 't10', None, 31)
Quad 23 ['+', 'm', 1, 't11']
Quad 24 ['=', 't11', None, 'm']
Quad 25 ['+', 1, 1, 't12']
Quad 26 ['*', 2, 4, 't13']
Quad 27 ['+', 't12', 't13', 't14']
Quad 28 ['=', 't14', None, 'e']
Quad 29 ['print', None, None, 'm']
8
```

```
Quad 30 ['GOTO', None, None, 17]
Quad 17 ['!=', 'c', 3, 't6']
Quad 18 ['==', 'm', 2, 't7']
Quad 19 ['AND', 't6', 't7', 't8']
Quad 20 ['<', 'm', 10, 't9']
Quad 21 ['OR', 't8', 't9', 't10']
Quad 22 ('GOTOF', 't10', None, 31)
Quad 23 ['+', 'm', 1, 't11']
Quad 24 ['=', 't11', None, 'm']
Quad 25 ['+', 1, 1, 't12']
Quad 26 ['*', 2, 4, 't13']
Quad 27 ['+', 't12', 't13', 't14']
Quad 28 ['=', 't14', None, 'e']
Quad 29 ['print', None, None, 'm']
9
Quad 30 ['GOTO', None, None, 17]
Quad 17 ['!=', 'c', 3, 't6']
Quad 18 ['==', 'm', 2, 't7']
Quad 19 ['AND', 't6', 't7', 't8']
Quad 20 ['<', 'm', 10, 't9']
Quad 21 ['OR', 't8', 't9', 't10']
Quad 22 ('GOTOF', 't10', None, 31)
Quad 23 ['+', 'm', 1, 't11']
Quad 24 ['=', 't11', None, 'm']
Quad 25 ['+', 1, 1, 't12']
Quad 26 ['*', 2, 4, 't13']
Quad 27 ['+', 't12', 't13', 't14']
Quad 28 ['=', 't14', None, 'e']
Quad 29 ['print', None, None, 'm']
10
Quad 30 ['GOTO', None, None, 17]
Quad 17 ['!=', 'c', 3, 't6']
Quad 18 ['==', 'm', 2, 't7']
Quad 19 ['AND', 't6', 't7', 't8']
Quad 20 ['<', 'm', 10, 't9']
Quad 21 ['OR', 't8', 't9', 't10']
Quad 22 ('GOTOF', 't10', None, 31)
Quad 31 ['*', 2, 3, 't15']
Quad 32 ['+', 't15', 5, 't16']
Quad 33 ['+', 't16', 6, 't17']
Quad 34 ['=', 't17', None, 'c']
```

Resultados

7
8
9
10

Pruebarray.bixo

Código

```
program pruebarray;

main(){
    var int ma;
    var int c,d,e;
    var float n;
    ma=1+1;
    e=2+3+5;
    n=2.3*3.3*5;
    array m[5]=[2,4,6,8,10,12];
    e=2+2+3;
    print(e);
    var float kk;
    kk=3.3*5;
    n=2.3*5;
    array dos[2]=[0,1,2];
    mean(m);
    find(m,6);
    print(m);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][1]
1: [FUNCSTART][main][None][1]
2: [+][1][1][t1]
3: [=][t1][None][ma]
4: [+][2][3][t2]
5: [+][t2][5][t3]
6: [=][t3][None][e]
7: [*][2.3][3.3][t4]
8: [*][t4][5][t5]
9: [=][t5][None][n]
10: [ARRAY][m][5][None]
11: [ARRAYSTART][t7][16001][m]
12: [ARRAYFILL][m[0]][t7][m]
13: [ARRAYFILL][m[1]][t8][m]
14: [ARRAYFILL][m[2]][t9][m]
15: [ARRAYFILL][m[3]][t10][m]
16: [ARRAYFILL][m[4]][t11][m]
17: [ARRAYFILL][m[5]][t12][m]
18: [ARRAYEND][t12][16001][m]
19: [+][2][2][t13]
20: [+][t13][3][t14]
21: [=][t14][None][e]
22: [print][None][None][e]
23: [*][3.3][5][t15]
24: [=][t15][None][kk]
25: [*][2.3][5][t16]
26: [=][t16][None][n]
27: [ARRAY][dos][2][None]
28: [ARRAYSTART][t18][16003][dos]
29: [ARRAYFILL][dos[0]][t18][dos]
30: [ARRAYFILL][dos[1]][t19][dos]
31: [ARRAYFILL][dos[2]][t20][dos]
32: [ARRAYEND][t20][16003][dos]
33: [MEAN][m][None][None]
34: [FIND][m][6][None]
35: [print][None][None][m]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 1)
Quad 1 ['FUNCSTART', 'main', None, 1]
Quad 2 ['+', 1, 1, 't1']
Quad 3 ['=', 't1', None, 'ma']
Quad 4 ['+', 2, 3, 't2']
Quad 5 ['+', 't2', 5, 't3']
Quad 6 ['=', 't3', None, 'e']
Quad 7 ['*', 2.3, 3.3, 't4']
Quad 8 ['*', 't4', 5, 't5']
Quad 9 ['=', 't5', None, 'n']
Quad 10 ['ARRAY', 'm', 5, None]
Quad 11 ['ARRAYSTART', 't7', 16001, 'm']
Quad 12 ['ARRAYFILL', 'm[0]', 't7', 'm']
Quad 13 ['ARRAYFILL', 'm[1]', 't8', 'm']
Quad 14 ['ARRAYFILL', 'm[2]', 't9', 'm']
Quad 15 ['ARRAYFILL', 'm[3]', 't10', 'm']
Quad 16 ['ARRAYFILL', 'm[4]', 't11', 'm']
Quad 17 ['ARRAYFILL', 'm[5]', 't12', 'm']
Quad 18 ['ARRAYEND', 't12', 16001, 'm']
Quad 19 ['+', 2, 2, 't13']
Quad 20 ['+', 't13', 3, 't14']
Quad 21 ['=', 't14', None, 'e']
Quad 22 ['print', None, None, 'e']
7.0
Quad 23 ['*', 3.3, 5, 't15']
Quad 24 ['=', 't15', None, 'kk']
Quad 25 ['*', 2.3, 5, 't16']
Quad 26 ['=', 't16', None, 'n']
Quad 27 ['ARRAY', 'dos', 2, None]
Quad 28 ['ARRAYSTART', 't18', 16003, 'dos']
Quad 29 ['ARRAYFILL', 'dos[0]', 't18', 'dos']
Quad 30 ['ARRAYFILL', 'dos[1]', 't19', 'dos']
Quad 31 ['ARRAYFILL', 'dos[2]', 't20', 'dos']
Quad 32 ['ARRAYEND', 't20', 16003, 'dos']
Quad 33 ['MEAN', 'm', None, None]
Mean de m = 7.0
Quad 34 ['FIND', 'm', 6, None]
VALUE 6 AT INDEX: (array([2], dtype=int64),)
Quad 35 ['print', None, None, 'm']
[ 2  4  6  8 10 12]
```

Resultados

7
Mean de m = 7.0
VALUE 6 AT INDEX: array[2]
[2 4 6 8 10 12]

PruebaMatrix.bixo

Código

```
program pruebamatrix;
function int que(int a){
    var int c,d,e;
    var float n;
    e=2+3+5;
    n=2.3*3.3*5;
    return e;
}
main(){
    matrix m[3][3]=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16];
    var float kk;
    kk=3.3*5;
    matrix z[3][3]=[2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32];
    mmult(m,z);
    print(z);
    mean(m);
    sort(m);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][9]
1: [FUNCSTART][que][None][1]
2: [+][2][3][t1]
3: [+][t1][5][t2]
4: [=][t2][None][e]
5: [*][2.3][3.3][t3]
6: [*][t3][5][t4]
7: [=][t4][None][n]
8: [FUNCEND][que][None][e]
9: [FUNCSTART][main][None][9]
10: [MATRIX][m][3][3]
11: [MATRIXSTART][t2][16000][m]
12: [MATRIXFILL][m[0][0]][t2][m]
13: [MATRIXFILL][m[0][1]][t3][m]
14: [MATRIXFILL][m[0][2]][t4][m]
15: [MATRIXFILL][m[0][3]][t5][m]
16: [MATRIXFILL][m[1][0]][t6][m]
17: [MATRIXFILL][m[1][1]][t7][m]
18: [MATRIXFILL][m[1][2]][t8][m]
19: [MATRIXFILL][m[1][3]][t9][m]
20: [MATRIXFILL][m[2][0]][t10][m]
21: [MATRIXFILL][m[2][1]][t11][m]
22: [MATRIXFILL][m[2][2]][t12][m]
23: [MATRIXFILL][m[2][3]][t13][m]
24: [MATRIXFILL][m[3][0]][t14][m]
25: [MATRIXFILL][m[3][1]][t15][m]
26: [MATRIXFILL][m[3][2]][t16][m]
27: [MATRIXFILL][m[3][3]][t17][m]
28: [MATRIXEND][t17][16000][m]
29: [*][3.3][5][t18]
30: [=][t18][None][kk]
31: [MATRIX][z][3][3]
32: [MATRIXSTART][t20][16002][z]
33: [MATRIXFILL][z[0][0]][t20][z]
34: [MATRIXFILL][z[0][1]][t21][z]
35: [MATRIXFILL][z[0][2]][t22][z]
36: [MATRIXFILL][z[0][3]][t23][z]
37: [MATRIXFILL][z[1][0]][t24][z]
38: [MATRIXFILL][z[1][1]][t25][z]
39: [MATRIXFILL][z[1][2]][t26][z]
40: [MATRIXFILL][z[1][3]][t27][z]
41: [MATRIXFILL][z[2][0]][t28][z]
42: [MATRIXFILL][z[2][1]][t29][z]
43: [MATRIXFILL][z[2][2]][t30][z]
44: [MATRIXFILL][z[2][3]][t31][z]
45: [MATRIXFILL][z[3][0]][t32][z]
46: [MATRIXFILL][z[3][1]][t33][z]
47: [MATRIXFILL][z[3][2]][t34][z]
48: [MATRIXFILL][z[3][3]][t35][z]
49: [MATRIXEND][t35][16002][z]
50: [MATRIXMULT][m][z][None]
51: [print][None][None][z]
52: [MEAN][m][None][None]
53: [SORT][m][None][None]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 9)
Quad 9 ['FUNCSTART', 'main', None, 9]
Quad 10 ['MATRIX', 'm', 3, 3]
Quad 11 ['MATRIXSTART', 't2', 16000, 'm']
Quad 12 ['MATRIXFILL', 'm[0][0]', 't2', 'm']
Quad 13 ['MATRIXFILL', 'm[0][1]', 't3', 'm']
Quad 14 ['MATRIXFILL', 'm[0][2]', 't4', 'm']
Quad 15 ['MATRIXFILL', 'm[0][3]', 't5', 'm']
Quad 16 ['MATRIXFILL', 'm[1][0]', 't6', 'm']
Quad 17 ['MATRIXFILL', 'm[1][1]', 't7', 'm']
Quad 18 ['MATRIXFILL', 'm[1][2]', 't8', 'm']
Quad 19 ['MATRIXFILL', 'm[1][3]', 't9', 'm']
Quad 20 ['MATRIXFILL', 'm[2][0]', 't10', 'm']
Quad 24 ['MATRIXFILL', 'm[3][0]', 't14', 'm']
Quad 25 ['MATRIXFILL', 'm[3][1]', 't15', 'm']
Quad 26 ['MATRIXFILL', 'm[3][2]', 't16', 'm']
Quad 27 ['MATRIXFILL', 'm[3][3]', 't17', 'm']
Quad 28 ['MATRIXEND', 't17', 16000, 'm']
Quad 29 ['*', 3.3, 5, 't18']
Quad 30 ['=', 't18', None, 'kk']
Quad 31 ['MATRIX', 'z', 3, 3]
Quad 32 ['MATRIXSTART', 't20', 16002, 'z']
Quad 33 ['MATRIXFILL', 'z[0][0]', 't20', 'z']
Quad 34 ['MATRIXFILL', 'z[0][1]', 't21', 'z']
Quad 35 ['MATRIXFILL', 'z[0][2]', 't22', 'z']
Quad 36 ['MATRIXFILL', 'z[0][3]', 't23', 'z']
Quad 37 ['MATRIXFILL', 'z[1][0]', 't24', 'z']
Quad 38 ['MATRIXFILL', 'z[1][1]', 't25', 'z']
Quad 39 ['MATRIXFILL', 'z[1][2]', 't26', 'z']
Quad 40 ['MATRIXFILL', 'z[1][3]', 't27', 'z']
Quad 41 ['MATRIXFILL', 'z[2][0]', 't28', 'z']
Quad 42 ['MATRIXFILL', 'z[2][1]', 't29', 'z']
Quad 43 ['MATRIXFILL', 'z[2][2]', 't30', 'z']
Quad 44 ['MATRIXFILL', 'z[2][3]', 't31', 'z']
Quad 45 ['MATRIXFILL', 'z[3][0]', 't32', 'z']
Quad 46 ['MATRIXFILL', 'z[3][1]', 't33', 'z']
Quad 47 ['MATRIXFILL', 'z[3][2]', 't34', 'z']
Quad 48 ['MATRIXFILL', 'z[3][3]', 't35', 'z']
Quad 49 ['MATRIXEND', 't35', 16002, 'z']
Quad 50 ['MATRIXMULT', 'm', 'z', None]
MULTIPLICACION DE m * z ES = [[ 180  200  220  240]
 [ 404  456  508  560]
 [ 628  712  796  880]
 [ 852  968 1084 1200]]
Quad 51 ['print', None, None, 'z']
[[ 2  4  6  8]
 [10 12 14 16]
 [18 20 22 24]
 [26 28 30 32]]
Quad 52 ['MEAN', 'm', None, None]
Mean de m = 8.5
Quad 53 ['SORT', 'm', None, None]
Sorted matrix m = [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Resultados

```
MULTIPLICACIÓN DE m * z ES =
[[ 180  200  220  240]
 [ 404  456  508  560]
 [ 628  712  796  880]
 [ 852  968 1084 1200]]


[[ 2  4  6  8]
 [10 12 14 16]
 [18 20 22 24]
 [26 28 30 32]]

Mean de m = 8.5

Sorted matrix m =
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

Fibonacci.bixo

Código

```
program f;
function void f() {
    var int c;
}

main() {
    var int n, result;
    var int a,b,i,temp;
    a=1-1;
    b=2-1;
    i=1+1;
    n = 7+1;
    while (i < n) {
        temp = a+b;
        a=b;
        b=temp;
        i=i+1;
    };
```

```
    print(b);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][3]
1: [FUNCSTART][f][None][1]
2: [FUNCEND][f][None][None]
3: [FUNCSTART][main][None][3]
4: [-][1][1][t1]
5: [=][t1][None][a]
6: [-][2][1][t2]
7: [=][t2][None][b]
8: [+][1][1][t3]
9: [=][t3][None][i]
10: [+][7][1][t4]
11: [=][t4][None][n]
12: [<][i][n][t5]
13: [GOTOF][t5][None][21]
14: [+][a][b][t6]
15: [=][t6][None][temp]
16: [=][b][None][a]
17: [=][temp][None][b]
18: [+][i][1][t7]
19: [=][t7][None][i]
20: [GOTO][None][None][12]
21: [print][None][None][b]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 3)
Quad 3 ['FUNCSTART', 'main', None, 3]
Quad 4 ['-', 1, 1, 't1']
Quad 5 ['=', 't1', None, 'a']
Quad 6 ['-', 2, 1, 't2']
Quad 7 ['=', 't2', None, 'b']
Quad 8 ['+', 1, 1, 't3']
Quad 9 ['=', 't3', None, 'i']
Quad 10 ['+', 7, 1, 't4']
Quad 11 ['=', 't4', None, 'n']
Quad 12 ['<', 'i', 'n', 't5']
Quad 13 ('GOTOF', 't5', None, 21)
Quad 14 ['+', 'a', 'b', 't6']
Quad 15 ['=', 't6', None, 'temp']
Quad 16 ['=', 'b', None, 'a']
Quad 17 ['=', 'temp', None, 'b']
Quad 18 ['+', 'i', 1, 't7']
Quad 19 ['=', 't7', None, 'i']
Quad 20 ['GOTO', None, None, 12]
Quad 12 ['<', 'i', 'n', 't5']
Quad 13 ('GOTOF', 't5', None, 21)
Quad 13 ('GOTOF', 't5', None, 21)
Quad 14 ['+', 'a', 'b', 't6']
Quad 15 ['=', 't6', None, 'temp']
Quad 16 ['=', 'b', None, 'a']
Quad 17 ['=', 'temp', None, 'b']
Quad 18 ['+', 'i', 1, 't7']
Quad 19 ['=', 't7', None, 'i']
Quad 20 ['GOTO', None, None, 12]
Quad 12 ['<', 'i', 'n', 't5']
Quad 13 ('GOTOF', 't5', None, 21)
Quad 14 ['+', 'a', 'b', 't6']
Quad 15 ['=', 't6', None, 'temp']
Quad 16 ['=', 'b', None, 'a']
Quad 17 ['=', 'temp', None, 'b']
Quad 18 ['+', 'i', 1, 't7']
Quad 19 ['=', 't7', None, 'i']
Quad 20 ['GOTO', None, None, 12]
Quad 12 ['<', 'i', 'n', 't5']
Quad 13 ('GOTOF', 't5', None, 21)
Quad 21 ['print', None, None, 'b']
13
```

Resultados

13

Factorial.bixo

Código

```
program fact;
function void f() {
    var int c;
}

main() {
    var int n, result;
    n = 4+1;
    result=2-1;
    while (n > 1) {
        result= result*n;
        n= n-1;
    };
    print(result);
}
```

Cuádruplos

```
Cuadruplos:  0: [GOTO][None][None][3]
1: [FUNCSTART][f][None][1]
2: [FUNCEND][f][None][None]
3: [FUNCSTART][main][None][3]
4: [+][4][1][t1]
5: [=][t1][None][n]
6: [-][2][1][t2]
7: [=][t2][None][result]
8: [>][n][1][t3]
9: [GOTOF][t3][None][15]
10: [*][result][n][t4]
11: [=][t4][None][result]
12: [-][n][1][t5]
13: [=][t5][None][n]
14: [GOTO][None][None][8]
15: [print][None][None][result]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 3)
Quad 3 ['FUNCSTART', 'main', None, 3]
Quad 4 ['+', 4, 1, 't1']
Quad 5 ['=', 't1', None, 'n']
Quad 6 ['-', 2, 1, 't2']
Quad 7 ['=', 't2', None, 'result']
Quad 8 ['>', 'n', 1, 't3']
Quad 9 ('GOTOF', 't3', None, 15)
Quad 10 ['*', 'result', 'n', 't4']
Quad 11 ['=', 't4', None, 'result']
Quad 12 ['-', 'n', 1, 't5']
Quad 13 ['=', 't5', None, 'n']
Quad 14 ['GOTO', None, None, 8]
Quad 8 ['>', 'n', 1, 't3']
Quad 9 ('GOTOF', 't3', None, 15)
Quad 10 ['*', 'result', 'n', 't4']
Quad 11 ['=', 't4', None, 'result']
Quad 12 ['-', 'n', 1, 't5']
Quad 13 ['=', 't5', None, 'n']
Quad 14 ['GOTO', None, None, 8]
Quad 8 ['>', 'n', 1, 't3']
Quad 9 ('GOTOF', 't3', None, 15)
Quad 10 ['*', 'result', 'n', 't4']
Quad 11 ['=', 't4', None, 'result']
Quad 12 ['-', 'n', 1, 't5']
Quad 13 ['=', 't5', None, 'n']
Quad 14 ['GOTO', None, None, 8]
Quad 8 ['>', 'n', 1, 't3']
Quad 9 ('GOTOF', 't3', None, 15)
Quad 10 ['*', 'result', 'n', 't4']
Quad 11 ['=', 't4', None, 'result']
Quad 12 ['-', 'n', 1, 't5']
Quad 13 ['=', 't5', None, 'n']
Quad 14 ['GOTO', None, None, 8]
Quad 8 ['>', 'n', 1, 't3']
Quad 9 ('GOTOF', 't3', None, 15)
Quad 15 ['print', None, None, 'result']
120
```

Resultados

120

Pruebafuncesp.bixo

Código

program pruebafuncesp;
function int que(int a){

```
    var int c,d,e;
    var float n;
    e=2+3+5;
    n=2.3*3.3*5;
    return e;
}
main(){
    array a[5]=[0,8,15,22,30,38];
    array m[5]=[32,46,59,72,86,100];
    var float kk;
    kk=3.3*5;
    print(a);
    layers(units=1);
    sequential();
    compile(0.1);
    fit(a,m, epochs = 1000);
    plot();
    predict(220.0);
    factorial(5);
    fibonacci(10);
    mean(m);
}
```

Cuádruplos

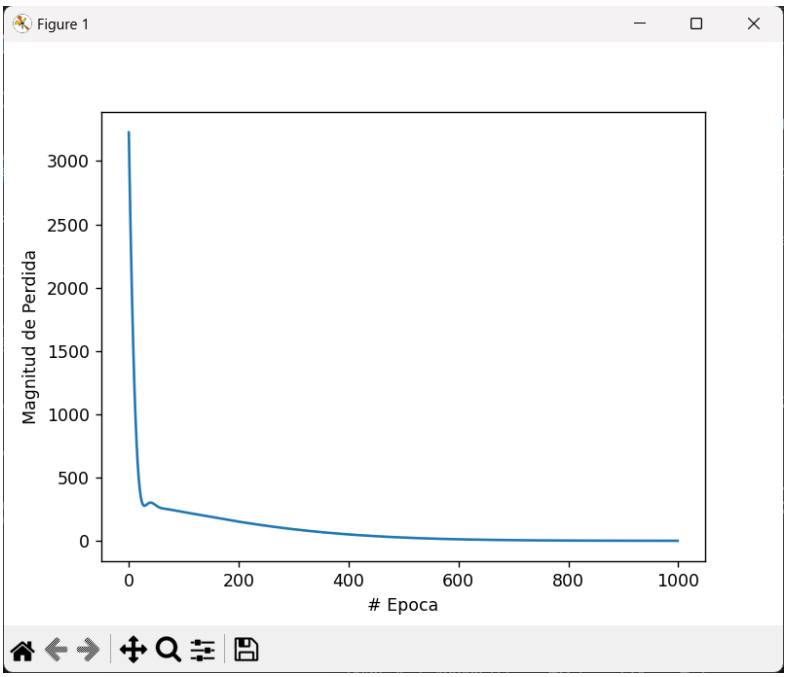
```
Cuadruplos:  0: [GOTO][None][None][9]
1: [FUNCSTART][que][None][1]
2: [+][2][3][t1]
3: [+][t1][5][t2]
4: [=][t2][None][e]
5: [*][2.3][3.3][t3]
6: [*][t3][5][t4]
7: [=][t4][None][n]
8: [FUNCEND][que][None][e]
9: [FUNCSTART][main][None][9]
10: [ARRAY][a][5][None]
11: [ARRAYSTART][t2][16000][a]
12: [ARRAYFILL][a[0]][t2][a]
13: [ARRAYFILL][a[1]][t3][a]
14: [ARRAYFILL][a[2]][t4][a]
15: [ARRAYFILL][a[3]][t5][a]
16: [ARRAYFILL][a[4]][t6][a]
17: [ARRAYFILL][a[5]][t7][a]
18: [ARRAYEND][t7][16000][a]
19: [ARRAY][m][5][None]
20: [ARRAYSTART][t9][16001][m]
21: [ARRAYFILL][m[0]][t9][m]
22: [ARRAYFILL][m[1]][t10][m]
23: [ARRAYFILL][m[2]][t11][m]
24: [ARRAYFILL][m[3]][t12][m]
25: [ARRAYFILL][m[4]][t13][m]
26: [ARRAYFILL][m[5]][t14][m]
27: [ARRAYEND][t14][16001][m]
28: [*][3.3][5][t15]
29: [=][t15][None][kk]
30: [print][None][None][a]
31: [LAYERS][1][None][None]
32: [SEQUENTIAL][None][None][None]
33: [COMPILE][0.1][None][None]
34: [FIT][a][m][1000]
35: [PREDICT][220.0][None][None]
36: [FACTORIAL][5][None][None]
37: [FIBONACCI][10][None][None]
38: [MEAN][m][None][None]
```

Orden de Ejecución de Cuádruplos

```
Quad 0 ('GOTO', None, None, 9)
Quad 9 ['FUNCSTART', 'main', None, 9]
Quad 10 ['ARRAY', 'a', 5, None]
Quad 11 ['ARRAYSTART', 't2', 16000, 'a']
Quad 12 ['ARRAYFILL', 'a[0]', 't2', 'a']
Quad 13 ['ARRAYFILL', 'a[1]', 't3', 'a']
Quad 14 ['ARRAYFILL', 'a[2]', 't4', 'a']
Quad 15 ['ARRAYFILL', 'a[3]', 't5', 'a']
Quad 16 ['ARRAYFILL', 'a[4]', 't6', 'a']
Quad 17 ['ARRAYFILL', 'a[5]', 't7', 'a']
Quad 18 ['ARRAYEND', 't7', 16000, 'a']
Quad 19 ['ARRAY', 'm', 5, None]
Quad 20 ['ARRAYSTART', 't9', 16001, 'm']
Quad 21 ['ARRAYFILL', 'm[0]', 't9', 'm']
Quad 30 ['print', None, None, 'a']
[ 0  8 15 22 30 38]
Quad 31 ['LAYERS', 1, None, None]
Quad 32 ['SEQUENTIAL', None, None, None]
Quad 33 ['COMPILE', 0.1, None, None]
Quad 34 ['FIT', 'a', 'm', 1000]
Comenzando entrenamiento...
Modelo entrenado!
Quad 35 ['PREDICT', 220.0, None, None]
Hagamos una predicción!
1/1 [==============================] - 0s 72ms/step
El resultado es: [[460.4257]]
Quad 36 ['GETWEIGHTS', None, None, None]
Variables internas del modelo
[array([[1.9686539]], dtype=float32), array([27.32184], dtype=float32)]
Quad 37 ['FACTORIAL', 5, None, None]
FACTORIAL DE 5 = 120
Quad 38 ['FIBONACCI', 10, None, None]
0
1
1
2
3
5
8
13
21
34
Quad 39 ['MEAN', 'm', None, None]
Mean de m = 65.83333333333333
```

Resultados

[0 8 15 22 30 38]
Comenzando entrenamiento...
Modelo entrenado!



Hagamos una predicción!
1/1 [==============================] - 0s 74ms/step
El resultado es: [[460.4257]]
Variables internas del modelo
[array([[1.8166671]], dtype=float32), array([31.459696], dtype=float32)]
FACTORIAL DE 5 = 120
0
1
1
2
3
5
8
13
21
34
Mean de m = 65.8333333333