Equipo 2 Jueves. Lenguaje BABAC

La descripción del lenguaje con el que vamos a trabajar tiene las siguientes características:

* Es fuertemente tipado.
* Los tipos de datos son: entero, real, carácter y booleano.
* Tiene una sentencia de asignación para todos los tipos de expresiones.
* Tendrá operadores lógicos.
* Las operaciones de entrada y salida serán a través del teclado y pantalla respectivamente, además el operador de entrada permite leer sobre una lista de identificadores y el operador de salida permite escribir sobre una lista de expresiones y/o constantes de tipo cadena.
* Estructuras de control:
  + si-sino
  + mientras
* En cualquier parte del programa se podrán definir bloques y la declaración de variables se realiza al comienzo del bloque delimitado con marcas de inicio y de fin.
* El lenguaje distingue entre mayúsculas y minúsculas.

Características concretas de nuestro lenguaje (BABAC)

* Está basado en el lenguaje C.
* Las palabras reservadas deben estar en castellano.
* Contiene una estructura de datos tipo array 1D y 2D que puede contener datos de tipo entero, real, carácter o booleano. Los métodos que contiene dicha estructura son:
  + De acceso a un elemento.
  + Producto de elemento a elemento.
  + Suma de elemento a elemento.
  + Resta de elemento a elemento.
  + Producto externo array por un escalar.
  + Producto de matrices C=A\*B.
* Los subprogramas deben ser definidos como funciones.
* La estructura de control a desarrollar es “repeat-until” (repetir\_hasta)

BNF

<programa> ::= <cabecera\_programa> <bloque>

<cabecera\_programa> ::= '**principal**'

<bloque> ::= <inicio\_bloque> <declar\_variables\_locales> <declar\_subprogs> <sentencias> <fin\_bloque>

<inicio\_bloque> ::= '**{**'

<declar\_variables\_locales> ::= <ini\_declar\_variables> <variables\_locales> <fin\_declar\_variables>

<declar\_subprogs> ::= <declar\_subprogs> <declar\_subprog>

<sentencias> ::= <sentencias> <sentencia>

| <sentencia>

<fin\_bloque> ::= '**}**'

<ini\_declar\_variables> ::= '**var**'

<variables\_locales> ::= <variables\_locales> <cuerpo\_declar\_variables>

| <cuerpo\_declar\_variables>

<fin\_declar\_variables> ::= '**finvar**'

<declar\_subprog> ::= <cabecera\_subprograma> <bloque>

<sentencia> ::= <bloque>

| <asignacion>

| <si>

| <mientras>

| <entrada>

| <salida>

| <devolver>

| <repetir\_hasta>

<cuerpo\_declar\_variables> ::= <tipo\_basico> <lista\_variables> '**;**'

<cabecera\_subprograma> ::= <tipo\_basico> <variable '**(**' <lista\_parametros> '**)**'

| <tipo\_basico> <identificador> '**(**' '**)**'

<asignacion> ::= <var\_exp> **'='** <expresion> '**;**'

<variable> ::= <identificador>

| <identificador> <declar\_matriz>

<var\_exp> ::= <identificador>

| <identificador> **'['** <expresion> **']'**

| <identificador> **'['** <expresion> **';'** <expresion> **']'**

<si> ::= '**si**' '**(**' <expresion> '**)**' <sentencia> '**sino**' <sentencia>

| '**si**' '**(**' <expresion> '**)**' <sentencia>

<mientras> ::= '**mientras**' '**(**' <expresion> '**)**' <sentencia>

<entrada> ::= '**leer**' <lista\_variables> '**;**'

<salida> ::= '**escribir**' <lista\_expresiones\_cad> '**;**'

<devolver> ::= '**devolver**' <expresion> '**;**'

<repetir\_hasta> ::= '**repetir**' <sentencia> '**hasta**' '(' <expresion> ')' '**;**'

<lista\_variables> ::= <lista\_variables> **','** <variable>

| <variable>

<identificador> ::= ([a-zA-Z\_] [0-9a-zA-Z\_]\*)

<lista\_parametros> ::= <lista\_parametros> '**,**' <tipo\_basico> <variable>

| <tipo\_basico> <variable>

<expresion> ::= '**(**' <expresion> '**)**'

| <op\_unario> <expresion>

| <expresion> <op\_binario> <expresion>

| <variable>

| <constante>

| <llamada\_funcion>

<lista\_expresiones> ::= <expresion> '**,**' <lista\_expresiones>

| <expresion>

<lista\_expresiones\_cad> ::= <lista\_expresiones\_cad> '**,**' <exp\_cad>

| <exp\_cad>

<exp\_cad> ::= <expresion> | <constante\_cadena>

<tipo\_basico> ::= '**caracter**'

| '**real**'

| '**entero**'

| '**logico**'

<declar\_matriz> ::= '**[**' <constante\_entera> '**]**'

| '**[**' <constante\_entera> ' **,**' <constante\_entera> '**]**'

<op\_unario> ::= '**+**'

| '**-**'

| '**!**'

<op\_binario> ::= '**+**'

| '**-**'

| '**\***'

| '**/**'

| '**==**'

| '**!=**'

| '**<=**'

| '**>=**'

| '**<**'

| '**>**'

| '**&&**'

| '**||**'

| '**^**'

<constante> ::= <constante\_entera>

| <constante\_matriz>

| <constante\_logica>

| <constante\_real>

| <constante\_caracter>

<llamada\_funcion> ::= <identificador> '**(**' <lista\_expresiones> '**)**' '**;**'

| <identificador '**(**' '**)**' '**;**'

<constante\_matriz> ::= '**{**' <auxiliar\_matriz> '**}**'

<auxiliar\_matriz> ::= <agregado\_entero>

| <agregado\_real>

| <agregado\_logico>

| <agregado\_caracter>

<agregado\_entero> ::= '**{**' <fila\_agregado\_entero> '**}**'

<fila\_agregado\_entero> ::= <fila\_agregado\_entero> '**;**' <lista\_entera>

| <lista\_entera>

<lista\_entera> ::= <lista\_entera> '**,**' <constante\_entera>

| <constante\_entera>

<agregado\_real> ::= '**{**' <fila\_agregado\_real> '**}**'

<fila\_agregado\_real> ::= <fila\_agregado\_real> '**;**' <lista\_real>

| <lista\_real>

<lista\_real> ::= <lista\_real> '**,**' <constante\_real>

| <constante\_real>

<agregado\_logico> ::= '**{**' <fila\_agregado\_logico> '**}**'

<fila\_agregado\_logico> ::= <fila\_agregado\_logico> '**;**' <lista\_logica>

| <lista\_logica>

<lista\_logico> ::= <lista\_logica> '**,**' <constante\_logico>

| <constante\_logica>

<agregado\_caracter> ::= '**{**' <fila\_agregado\_caracter> '**}**'

<fila\_agregado\_caracter> ::= <fila\_agregado\_caracter> '**;**' <lista\_caracter>

| <lista\_caracter>

<lista\_caracter> ::= <lista\_caracter> '**,**' <constante\_caracter>

| <constante\_caracter>

<constante\_entera> ::= [0-9]+

<constante\_cadena> ::= '**"**' [^"]+ '**"**'

<constante\_logica> ::= '**verdadero**'

| '**falso**'

<constante\_real> ::= <constante\_entera> '**.**' <constante\_entera>

<constante\_caracter> ::= '**'**' [^'] '**'**'