WHILE

ciclo:

    WHILE {apilar(\_tercetosCreados); crearTerceto(“ET”,\_;\_) } ID IN {apilar(tercetosCreados);avanzar}COR\_A lista COR\_C DO programa {crearTerceto(“BI”;\_;\_)

Desapilar(\_nroTope);escribirENtercetoPrevio(\_nroTope,\_tercetosCreados)

ENDWHILE

//sacarDe cola y poner en una auxiliar y despues apuntar la pirnicipal a la auxiliar y listo todos feliz menos el profe

lista:

    factor                                              \_contList=1;

    | lista COMA factor   \_contList++;

expresion:

    expresion OP\_SUMA termino         crearTerceto( “OP\_SUMA”;Eind;Tind)

    | expresion OP\_RESTA termino crearTerceto( “OP\_RESTA”;Eomd;Tind)

    | termino                     Eind=Tind

    |OP\_RESTA expresion %prec MENOS\_UNARIO   crearTerceto( “OP\_RESTA”,Eind,”\_”)

    |longitud   Eind=LongInd

longitud:

    LONG PAR\_A lista PAR\_C ListaInd = crearTerceto(\_cont;\_;\_);

termino:

    termino OP\_MULT factor         crearTerceto( “OP\_MULT”;Tind;Find)

    | termino OP\_DIV factor          crearTerceto( “OP\_DIV”;Tind;Find)

    | factor Tind= Find;

factor:

    PAR\_A expresion PAR\_C     Find= Eind

    | CONST\_REAL               crearTerceto(yytext,\_,\_);

    | ID

    | CONST\_ENT crearTerceto(yytext,\_,\_);

comparador:

    OP\_MAY strcpy(\_comparador ,BLE)

    | OP\_MEN         strcpy(\_comparador ,BGE)

    | OP\_MENIGU     strcpy(\_comparador ,BGT)

    | OP\_MAYIGU     strcpy(\_comparador ,BLT)

    | OP\_IGUAL       strcpy(\_comparador ,BNE)

    | OP\_DIF         strcpy(\_comparador ,BNQ)

seleccion:

    IF condicion {crearTerceto(CMP,\_condicion ;\_); apilarIndiceActual(tercetosCreados),avanzar()} THEN programa{desarpilarIndice(\_nroTerceto); escribirEnceldaX(tercetosCreados,nroTerceto)} ELSE programa {desapilarIndice(\_nroTecero); escribirEnCeldaX(tercetosCreados,nroTerceto)} ENDIF

    | IF condicion {crearTerceto(CMP,\_condicion ;\_); apilarIndiceActual(tercetosCreados),avanzar()} THEN programa {desarpilarIndice(\_nroTerceto); escribirEnceldaX(tercetosCreados,nroTerceto)} ENDIF

condicion:

    comparacion CondicionInd = ComparacioInd

    | condicion AND comparacion                      crearTerceto(AND,CondicionInd, ComparacioInd)

    | condicion OR comparacion                          CrearTerceto(OR, CondicionInd, ComparacioInd)

    | PAR\_A NOT condicion PAR\_C AND comparacion   crearTerceto();

    | PAR\_A  NOT condicion PAR\_C OR comparacion crearTerceto();

Preguntar al profe

comparacion:

    expresion comparador expresión crearTerceto(\_comparador;Eind;Eind)//sacarDePila los 2 primeros(“””””CREO”””””)

entrada:

    GET ID ¿???????????

salida:

    DISPLAY CONST\_STR     ¿???????????????

declaracion:

    DECVAR  dec ENDDECVAR declaracionInd= decInd

dec:

    dec DIM COR\_A listavar COR\_C AS COR\_A listatipodato COR\_ decInd= crearTerceto(AS,ListaVar,tipoDato)

    | DIM COR\_A listavar COR\_C AS COR\_A listatipodato COR\_C   decInd= crearTerceto(AS,ListaVar,tipoDato)

listavar:

    ID                             ListaVar =  crearTerceto(yytex;\_;\_)

    | listavar COMA ID ListaVar=  crearTerceto(ytext;\_;\_)

listatipodato:

    tipodato                        ¿??????????????

    | listatipodato COMA ¿?????

tipodato:

    INTEGER                         {\_tipoDato=”INTEGER” }

    | FLOAT                         {\_tipoDato= “FLOAT”}

    | STRING   {tipoDato=”STRING”}

sentencia:

     declaracion                     SentenciaInd = declaracionInd

    |asignacion                       SentenciaInd=asignacionInd

    |ciclo                           SentenciaInd=cicloInd

    |seleccion                       SentenciaInd = seleccionInd

    |salida                           SentenciaInd = salidaInd

    |entrada                         SentenciaInd = entradaInd

    |longitud SentenciaInd = longitudInd

programa:

    sentencia                       programaInd = sentenciaInd

    | programa sentencia  programa

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

WHILE var1 IN [ a , c , d]

DO

var1 := var2 + 3

ENDWHILE

ETW \_aux = desapilar();

If (var1 == Aux)

{

If()

Var1=var2+3;

} go to ETW

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [2](var1;\_;\_)  [3]( a;\_;\_ )  [4](c;\_;\_)  [5](d;\_;\_)  [6] (ETW;\_;\_)  //pilaVacia  [7](:=, @aux, Lind)  [8]( CMP,[2], @aux)  [9] (BNE,\_,[ 6 ] )  [10] (3,\_,\_)  [11] (var2,\_,\_)  [12](+,\_,\_)  [13] | Aind = |  |  | ListaInd |  |  |  | PILANTeceto |
| Pind |  |  |  |  |  |  |  |
| Lind = |
| Eind =2 |
| Tind= |
| Find = |