

INF265 2020-2
Teoría de compiladores
PREGUNTA 1

Considerando las siguientes palabras reservadas, tokens y gramática:

Palabra reservada	token
if	SI
else	SINO
while	MIENTRAS
int	ENTERO
double	DECIMAL
printf	MUESTRA
read	LECTURA

```

programa: listainst; /*R1*/
listainst: instr listainst /*R2*/
          | instr ; /*R3*/
instr : asignacion ';' /*R4*/
       | selectiva /*R5*/
       | iterativa /*R6*/
       | declaracion ';' /*R7*/
       | impresion ';' /*R8*/
       | lectura ';;' /*R9*/
asignacion: ID IGUAL expr ; /*R10*/
expr : expr '+' term ; /*R11*/
expr : expr '-' term ; /*R12*/
expr : term ; /*R13*/
term : term '*' factor /*R14*/
      | factor ; /*R15*/
factor : NUMBER /*R16*/
        | ID /*R17*/
        | '(' expr ')' ; /*R18*/
impresion: MUESTRA ID ; /*R19*/
tipodato: ENTERO /*R20*/
          | DECIMAL ; /*R21*/
declaracion: tipodato listaidentificador ; /*R22*/
listaidentificador: listaidentificador ',' ID /*R23*/
                  | ID ; /*R24*/
cond: expr '<' expr /*R25*/
      | expr '>' expr /*R26*/
      | expr IGUAL IGUAL expr /*R27*/
      | expr '>' IGUAL expr; /*R28*/
bloque: '{' listainst '}' /*R29*/
        | instr ';' ; /*R30*/
lectura: LECTURA ID; /*R31*/
selectiva: SI '(' cond ')' bloque otro ; /*R32*/
otro: SINO bloque /*R33*/
      | ; /*R34*/
iterativa: MIENTRAS '(' cond ')' bloque ; /*R35*/

```

Se le pide elaborar el árbol de derivación de la siguiente cadena:
Recuerde que debe convertir la cadena a una secuencia de tokens y luego elaborar el árbol, para comprobar si la cadena es válida. Coloque en cada derivación el identificador de la regla que uso.

```

z=0;
while(z<20){
  if(z<10){
    printf z;
  }
  z=z+1;
}
}

```