

NOMBRE:

1.

2.

3.
$$\begin{array}{l} \mathbf{E} \rightarrow \mathbf{E} \text{ or } \mathbf{T} \\ \quad | \mathbf{T} \end{array}$$
$$\begin{array}{l} \mathbf{T} \rightarrow \mathbf{T} \text{ and } \mathbf{F} \\ \quad | \mathbf{F} \end{array}$$
$$\begin{array}{l} \mathbf{F} \rightarrow \mathbf{id} \\ \quad | (\mathbf{E}) \\ \quad | \mathbf{id}[\mathbf{E}] \end{array}$$

COMPLETA LA TABLA

Analiza la entrada: `id * id | id`

¿Es aceptada la entrada por la gramática?

5.

ESTADO	acción				goto
	id	-	+	\$	E
0	s3	s2			1
1			s4	ace	
2	s3	s2		p	5
3			r3	r3	
4	s3	s2			6
5			r2	r2	
6			r1	r1	

.
Analiza la secuencia - - **id + id - id**

Pila	Entrada	Acción
0	-- id + id - id \$	desplazar

¿Es aceptada la entrada por la gramática?

6.

1. $i = 1$
2. $j = 1$
3. $t1 = 10 * i$
4. $t2 = t1 + j$
5. $t3 = 8 * t2$
6. $t4 = t3 - 88$
7. $a[t4] = 0.0$
8. $j = j + 1$
9. if $j \leq 10$ goto (3)
10. $i = i + 1$
11. if $i \leq 10$ goto (2)
12. $i = 1$
13. $t5 = i - 1$
14. $t6 = 88 * t5$
15. $a[t6] = 1.0$
16. $i = i + 1$
17. if $i \leq 10$ goto (13)

7.

Instrucción	Código generado	Descriptor de registros	Descriptor de direcciones
		R0: vacío R1:b	a: en memoria c: en memoria d: en memoria b: en R1
1) c := b - a (b,v,2) (c,v,2) (a,v,3)			
2) d := c - b (d,v,3) (c,m) (b,v,4)			
3) c := d - a (c,v,5) (d,v,5) (a,m)			
4) a := b (a,v,5) (b,v,5)			

8. Considera la cabecera siguiente para la especificación Bison parser.y:

```

struct expresionstruct {           %union {                               %type <expr> expr
string str ;                      string *str ;                       %type <str> stmts
vector<int> trues ;               expresionstruct *expr ;           %type <str> stmt
vector<int> falses ;             int number ;
};                                }

```

```
stmt : RWHILE M
```

```
expr RDO
```

```
    stmts M RENDWHILE
```

```
    { codigo.completarInstrucciones($3->trues,$5) ;
```

```
      codigo.completarInstrucciones($3->falses,$7) ;
```

```
      codigo.anadirInstruccion("goto");
```

```
      vector<int> tmp1 ; tmp1.push_back($7) ;
```

```
      codigo.completarInstrucciones(tmp1, $5) ;
```

```
    }
```

```
;
```

```
expr : expr TCEQ expr
```

```
    { expresionstruct *$$;
```

```
      codigo.anadirInstruccion("if " + $4->str + *$5 + $6->str + " goto") ;
```

```
      codigo.anadirInstruccion("goto") ;
```

```
      $$->trues.push_back(codigo.obtenRef()) ;
```

```
$$->falses.push_back(codigo.obtenRef()+1) ; }
```

9.