

Ejercicios de BNF  $\Leftrightarrow$  PAS

Nota: los ejercicios marcados con (\*) al principio están sacados del libro de la cátedra  
los ejercicios marcados con (°) al principio están basados en uno tomado en un final

1. (°) Dada la siguiente BNF:  
 $\langle \text{sentenciaAsignacion} \rangle ::= \langle \text{identificador} \rangle \text{ OpAsignacion } \langle \text{expresion} \rangle \text{ SgPuntuacion}$   
Construya una función ANSI C que implemente el PAS de esta producción

2. (°) Dada la siguiente BNF:  
 $\langle \text{FACTOR} \rangle ::= \text{PAREN\_IZQ } \langle \text{EXP} \rangle \text{ PAREN\_DER } \mid \text{NR0}$   
Construya una función ANSI C que implemente el PAS de esta producción

3. Dada la siguiente BNF:  
 $\langle \text{Exp} \rangle ::= \langle \text{Operando} \rangle \{ \text{SUM } \langle \text{Operando} \rangle \mid \text{MULT } \langle \text{Operando} \rangle \}$   
Construya una función ANSI C que implemente el PAS de esta producción

4. (°) Sea el siguiente PAS  

```
void PS(void) {  
    TOKEN tok = ProximoToken();  
    switch(tok) {  
        case A: Match(A); break;  
        case B: Match(B); break;  
        case C: Match(C); D(); Match(E); break;  
        default: ErrorSintactico(tok); break;  
    }  
}
```

Escriba las producciones que implementa dicho PAS.

5. (°) Sea el siguiente PAS  

```
void Sent1(void) {  
    Sent2();  
    while(1) {  
        switch(ProximoToken()) {  
            case SGN0: case OP_ESP: Sent2(); break;  
            default: return;  
        }  
    }  
}
```

Escriba las producciones que implementa dicho PAS. Debe utilizar recursividad a derecha

6. (°) Sea el siguiente PAS  

```
void Y(void) {  
    Match(H);  
    while(1) {  
        switch(ProximoToken()) {  
            case J: Match(J); break;  
            case Z: Match(Z); break;  
            default: return;  
        }  
    }  
}
```

Escriba las producciones que implementa dicho PAS. Debe utilizar recursividad a derecha