Parcial 02 Recuperatorio 01

1) Encuentre el conjunto Primero(S) para la siguiente gramática

```
S → XYZ          X → aM | ε      Y → ε | fM          Z → ε | ZYg
```

Solución

Primero(S) = {a, f, g, ε}

2) Dada la siguiente GIC encuentre una equivalente LL(1)

```
S → Sopa | rT          T → prEf | proF | pr
```

Solución

```
S → rTS'          S' → opaS' | ε
T → prT'          T' → Ef | oF | ε
```

3) Escriba la EBNF correspondiente a la siguiente PAS.

```
void Iteracion()
{
      TOKEN t = ProximoToken();
      for (; t == UNO || t == DOS; t = ProximoToken()) {
            Match(t);
            Caso();
      }
      Stop();
      Match(FIN);
}
```

Solución

```
<Iteracion> ::= { UNO <Caso> | DOS <Caso> } <Stop>  FIN
```

4) Dado el siguiente fragmento del estándar de C

```
inclusive-OR-expression:
      exclusive-OR-expression
      inclusive-OR-expression | exclusive-OR-expression

logical-AND-expression:
      inclusive-OR-expression
      logical-AND-expression && inclusive-OR-expression
```

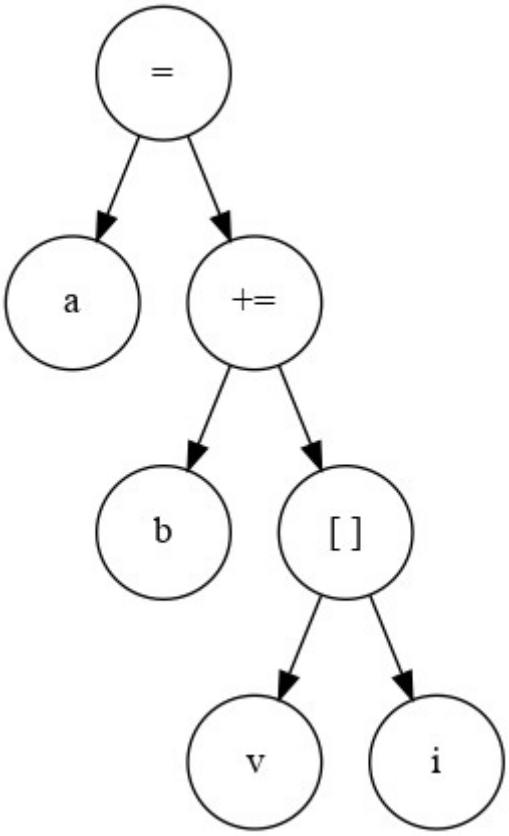
a) Indique, justificando, como asocia el operador **&&**.

De izquierda a derecha, porque la recursión es a izquierda

b) Indique, justificando, que operador tiene mayor precedencia entre **&&** y **|**.

El operador **|** tiene mayor precedencia que **&&** porque está más lejos del axioma.

5) Dibuje el árbol de la expresión en lenguaje C: `a = b += v[i]`



Dada la siguiente expresión en lenguaje C:

```
*v[p->a]
```

6) Haga las declaraciones necesarias para que sea semánticamente correcta y de tipo float.

```
float *v[5];
struct e {
      int a;
} *p;
```

7) Haga las declaraciones necesarias para que sea semánticamente incorrecta y explique el motivo.
Si declaro float `v`, entonces hay incompatibilidad con el operador `[]` que ya no tiene ningún operador puntero.

Dado el siguiente fuente

```
25    void recuperatorio(void) {
26        const int dim = 5;
27        int vec[dim] = {3, dim, 7};
28        int n;
29        for (int i =0; i < 10; i++) {
30            n=vec[i]++;
31            printf("vec[%d] = %d\n", i , vec[i]);
32        }
33    }
```

Conteste las siguientes preguntas.

8) A qué categoría sintáctica pertenece lo escrito en la línea 27?
Declaración/Definición.

9) De la lista de caracteres que serán devueltos con ungetc por el scanner en la línea 30.
`=, v, [, ]`

10) Indique el tipo de dato del último argumento de `printf` en línea 31. Justifique.
Es de tipo `int` ya que `vec` es arreglo de `int`.