Laboratorio de Procesadores de Lenguaje. Curso 2000-2001. 5 de Septiembre.
Tiempo: 1 hora 30 min. No se pueden usar libros ni apuntes.

Nombre del Alumno: **Grupo de Prácticas:**

1. (1pto.) Contesta a las siguientes cuestiones **justificando** tu respuesta.

   - Si se desea generar código, ¿cuál es la herramienta más adecuada: PCCTS ó FLEX/BISON?
   - ¿Qué mecanismos posee BISON para establecer la precedencia y asociatividad de operadores?
   - Comenta si la siguiente afirmación es verdadera o falsa: "PCCTS permite hacer uso de la notación EBNF para la especificación de la estructura léxica y sintáctica de un lenguaje".
   - Comenta si la siguiente afirmación es verdadera o falsa: "Dado que BISON genera una función por producción es fácil depurar un programa".

2. (2 ptos.) Una fábrica de muebles lanza al mercado su nueva colección de armarios que básicamente están compuestos de seis tipos diferentes de cajoneras y de tres tipos de estantes. Las cajoneras son de tres tamaños (*grande, mediana, pequeña*) y pueden constar de dos o tres cajones (lo que nos da los seis tipos diferentes de cajoneras). Estas se pueden combinar con un número indefinido de estantes que son de tamaño *grande, mediano, pequeño* o incluso no llevar ninguno. Las combinaciones de los estantes con las cajoneras se rigen por las siguientes reglas:

   - Se pueden juntar estantes grandes con pequeños siempre que uno grande sea sustituido por dos pequeños.
   - Los pequeños y los medianos no son intercambiables en ningún caso.
   - Se pueden poner tantos estantes como se desee siempre y cuando se correspondan en tamaño con los de las cajoneras.

   Se desea construir un programa informático que detecte automáticamente la secuencia en que llega el materia por la cinta de construcción de forma que se puedan elaborar todos los muebles que se deseen, asumiendo que siempre se podrá formar al menos uno. Se pide:

   Construir la gramática LL(1) que permita a dicho programa detectar todas las posibilidades anteriores.

3. (3 ptos.) Dada la siguiente gramática, implementa un pequeño programa en BISON que calcule el número de variables declaradas, imprima su nombre y el valor de las expresiones.

   `Programa` → `ListaVbles ListaSenten`
   `ListaVbles` → `Tipo` **id** `;` `ListaVbles` | ϵ
   `Tipo` → `int` | `real`
   `ListaSenten` → **id** `=` `Expr ;` `ListaSenten` | ϵ
   `Expr` → `Expr` + `Expr` | `Expr - Expr` | **num**

   Para la siguiente entrada indica el valor que tomarán los atributos definidos.

   ```
   int a;
   a=3+4;
   ```

4. (4 ptos.) Una pequeña calculadora nos permite convertir el resultado de nuestras operaciones aritméticas automáticamente de pesetas a euros. La gramática correspondiente es:

   `Entrada` → `Ecuacion ListaEcuaciones`
   `ListaEcuaciones` → `Ecuacion ListaEcuaciones` | ϵ
   `Ecuacion` → **id** `=` `Expr /` **EURO**
   `Expr` → **num** `ExprP`
   `ExprP` → + **num** `ExprP` | `-` **num** `ExprP` | ϵ
   Se pide:

   - Compactar la gramática y escribe la parte del código de PCCTS que se corresponde con la definición del analizador sintáctico `class MiParser`.
   - Construye el árbol sintáctico y dibújalo en notación PCCTS para la entrada:
     `id=3-4+8;`
   - Implementa la función que genera código dentro del PCCTS para la clase AST, con el fin de que contemple este tipo de instrucciones. El código generado debe estar en la notación de cuádruplos.