**Tecnológico de Monterrey, campus Guadalajara**

**División ingeniería**

**Dep. de ciencias computacionales**

Materia: Compiladores

Prof: José Jesús Ambriz Meza

Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fecha: \_\_\_\_\_\_\_\_\_\_\_

**Instrucciones: Lee atentamente las instrucciones de cada sección y contesta brevemente. Cualquier duda con el profesor.**

**Sección I**

Responde brevemente a las siguientes preguntas ( 50 pts )

1. Cuál de los elementos de una gramática es el que genera combinaciones de terminales y no terminales infinitas?
2. El lenguaje C es un compilador de 1 o de varias pasadas?
3. Cuál es la parte de un compilador del lenguaje C que encontraría el siguiente error:

if numero == 2

{

printf(“hola mundo”);

}

El error indicaría que faltan los paréntesis que rodean a la condición del if

1. Cuál es la parte de un compilador del lenguaje C que encontraría el siguiente error:

if (numero $= 2 )

{

printf(“hola mundo”);

}

(5) Cuál es el elemento de un lenguaje que no puede ser descrito en una gramática?

(6) Qué tipo de gramática es a la que pertenece el lenguaje C?

(7) Cuáles son las etapas del compilador que son válidas durante todas las etapas de la compilación?

(8) En qué etapa del proceso de compilación se genera el árbol sintáctico?

(9) El analizador sintáctico recursivo descendente para que tipo de estrategia es: LL o LR?

(10) Qué guarda la tabla de símbolos ?

**Sección 2**

Realiza los siguientes ejercicios.( 50 pts )

1. En la clase se analizó la gramática del proyecto final para while, if e iterate. Completa/corrige la siguiente gramática del SWITCH para el proyecto (15 pts)

<expression> ::= <call function> | <switch expression> | <if expression> | <while expression> | <iterate expression>

<switch expression> ::= "switch" “(“ <Case conditionals> “)” "{" <cases> "}"

<Case conditionals> ::= GET\_VALUE | GET\_COLOR

GET\_VALUE => devuelve el valor numérico de la carta que se tiene en la mano

GET\_COLOR => devuelve el color de la carta que se tiene en la mano

<cases> ::=

<case> ::= "case" "(" <number> ")" “:” "{" "}"

1. Usando la gramática del proyecto escribe los programas que resuelvan los siguientes “problemas” de programación: (35 pts )
2. Separa todos los naipes en 4 mazos ( 1 por cada mazo diferente) (5 pts )
3. Separa todos los naipes en 2 mazos ( 1 por cada color diferente) (5 pts )
4. Separa en 1 mazo todas las cartas que se encuentren en medio del primer par de corazones (10 pts )
5. Separa en 1 mazo todas las cartas que se encuentren en medio de un par de corazones ( 15 pts )