**Proyecto de Compiladores 1**

El proyecto se elabora de manera individual o en parejas, deberá de ser construido el ide en java y el resto del compilador en lenguaje puede ser libre, excluyendo c, c++ y java. El programa deberá de ser terminado en su totalidad ya que sirve como plataforma para el segundo curso.

**Las partes a elaborar son:**

1. Parte: IDE, Fecha de entrega 11 y 12 de Abril del 2023.

2. Parte: Analizador Léxico. Fecha de entrega 8 y 9 de Mayo 2023

3. Parte: Analizador Sintáctico. Fecha de entrega 29 y 30 de Mayo 2023

Recuerde que cada día hábil de retraso en la entrega de cada una de las fases causara una penalización del 10%, si llegara a un total de 0 en calificación de alguna de las fases, no lo excluye de su entrega correcta.

**Parte 2 Analizador Lexicográfico**

A continuación, se describe un lenguaje de programación denominado Tiny-extended, que incluye **opcionalmente** funciones. La especificación del lenguaje está dividida en tres secciones. En la primera parte se enumeran las convenciones léxicas del lenguaje, incluyendo una descripción de los tokens del lenguaje.

Recuerde que su programa deberá de ser capaz de leer un archivo (programa) de entrada desde la línea de comandos y generara como salida un archivo con la información de identificación de cada uno de los componentes léxicos del lenguaje, los errores ocurridos podrán ser enviado a otro archivo o a la salida estándar y recuperados por el IDE.

Convenciones léxicas (deberá realizar esta etapa implementando autómatas de estado finito):

* Las palabras clave o reservadas del lenguaje de programación son las siguientes, recuerde que las palabras clave o reservadas deben ser escritas en minúsculas:

main, if, then, else, end, do, while, repeat, until, cin, cout, real, int, boolean

* Los símbolos especiales son los siguientes:

+ - \* / % < <= > >= == != := ( ) { } // /\*\*/ ++ -- , ;

* Otros tokens son los identificadores y los números, definidos mediante las siguientes expresiones regulares, recuerde que hay distinción entre mayúsculas y minúsculas:

letra = a|..|z|A|..|Z

digito = 0|..|9

identificador = letra ( letra | digito |\_)\*

entero = (+|-)? digito+

real = (+|-)? digito+ (. digito+)?

* Los espacios en blanco se componen de blancos, retornos de línea y tabulaciones. El espacio en blanco es ignorado, excepto cuando deba separar identificadores, números y palabras reservadas, por ejemplo:

int variable1,x,y,z;

if(x==1)x=1;else x=0;end

* Los comentarios están encerrados entre anotaciones habituales del lenguaje C /\*…\*/ y comentarios en línea //. Los comentarios se pueden colocar en cualquier lugar donde pueda aparecer un espacio en blanco (es decir, los comentarios no pueden ser colocados dentro de los tokens) y para el caso de /\*…\*/ pueden incluir más de una línea. Los comentarios no pueden estar anidados.

Además de lo ya mencionado, el compilador debe ser capaz de pintar los tokens de diferente color, por ejemplo: palabras reservadas con azul, numero con rojo, comentarios con gris, etc.

Parte 3 Analizador sintáctico…