Descripción del problema

La primera fase de un compilador es el analizador léxico, el cual recibe como entrada un código fuente, para realizar un análisis e identificar aquellos elementos que pertenezcan al alfabeto definido, al igual que detectar los errores léxicos, la salida de este debe ser una lista de tokens, los cuales son una secuencia de componentes léxicos que siguen reglas léxicas expresadas mediante expresiones regulares.

En este caso las reglas léxicas se deben definir para el lenguaje de programación definido en clase. Dicho lenguaje tiene ocho distintas clases las cuales se enlistan en la siguiente tabla:

Clase	Descripción
0	Palabras reservadas
1	Identificadores (Solo letras, inician con mayúsculas y hasta 8 letras)
2	Símbolos especiales
3	Operadores de asignación
4	Operadores de relación
5	Operadores aritméticos
6	Constantes cadenas
7	Constantes numéricas
8	Constantes numéricas reales

Expresiones regulares usadas:

```
minuscula [a-z]
mayuscula [A-Z]
Letras ({minuscula}|{mayuscula}){0,7}
num [0-9]
rel DIF|IGL|MN|MNI|MY|MYI
punto \.
numEntero {num}+
numReal {num}+{punto}{num}+|{num}+{punto}{num}*|{num}*{punto}{num}+
exp (E|e)
signo (\+|-){0,1}
valExp {num}{1,2}

palRes Bul|Cadena|Cierto|Entero|Falso|Haz|Mientras|Para|Real|Si|Sino
iden {minuscula}{Letras}
simEsp (\(|\)|,|;|\[|\])
opAsig :=
opRel  {punto}{rel}{punto}
opArit (\+|-|\*|%|\/)
consCad  \".*\"
consNum {numEntero}
consNumReal {numReal}{exp}{signo}{valExp}|{numReal}
comentario \$\$.*
errorGeneral .
```

El analizador léxico también debe generar las tablas dinámicas y estáticas que se utilizarán en el programa, en este caso serán 3 tablas estáticas:

- Tabla de clases que debe tener el número de la clase, y su descripción
- Tabla de palabras reservadas que igualmente debe tener un número de identificador y la palabra reservada.
- Tabla de operadores relacionales, contiene dos campos: valor y operador relacional.

Tres tablas dinámicas:

- Tabla de símbolos donde se guardarán los identificadores, esta tabla contiene tres campos: Posición, identificador, y tipo este último vacío. Cada vez que se detecte un nuevo identificador se tiene que hacer una búsqueda.
- Tabla de cadenas, al llegar una cadena se debe guardar en esta tabla, la cual tiene dos campos: posición y cadena.
- Tabla de tokens, tiene dos espacios, clase a la que pertenece y el valor del token, este varía según la clase a la que pertenezca.

Análisis:

- Para el analizador léxico se deben definir las expresiones regulares para ello se transforman las especificaciones del lenguaje a expresiones regulares y posteriormente se llevan al lenguaje Lex.
- Una vez con todas las reglas se deben definir las tablas y métodos de búsqueda e inserción a emplearse.

Análisis:

Para el analizador léxico se deben definir las expresiones regulares para esto se deben transformar las especificaciones del lenguaje a expresiones regulares para posteriormente llevarlas a lenguaje Lex.

Una vez con todas las reglas se deben definir las tablas y métodos de búsqueda e inserción a emplearse. En este caso se usarán listas por lo que los métodos de búsqueda e inserción son para listas.

Actividades de cada integrante

Cuevas Salgado Carlos:

Diseño e implementación de las expresiones regulares.
Diseño de las pruebas de software.
Pruebas de software en las expresiones regulares, en las tablas, métodos de búsqueda e inserción.
Correcciones menores a partir de los resultados de las pruebas.

Chavez Galindo Lisset América:

Diseño e implementación de las tablas dinámicas/estáticas.
Método de búsqueda en las tablas estáticas basadas en arreglos.

Métodos de búsqueda, inserción e impresión en las tablas estáticas, basadas en listas ligadas.
Pruebas de software en las tablas, métodos de búsqueda e inserción.
Correcciones menores a partir de los resultados de las pruebas.

Diseño e implementación:

Las tablas estáticas se definirán mediante un arreglo de apuntador, de la siguiente forma char *tablaEstatica = {valor1, valor2, … , valorn }
Para hacer la búsqueda en la tabla estática se implementará una función donde se mandará la tabla estática y el valor, se hará una comparación para encontrar el valor buscado con los valores de la tabla estática, y se retornará la posición indicada en la tabla estática.

Para las tablas dinámicas se usarán listas ligadas, en donde se mandará el valor a guardar. Para cada tabla se definirá su propia estructura, tanto para el nodo de cada una como para su lista enlazada, ya que los valores que utilizan son diferentes para cada una. La búsqueda en las listas se hace mandando el token y recorriendo toda la lista buscando el valor, si se encuentra en la lista no se guardará en caso contrario se almacenará al final de la lista.

Funcionamiento del programa, se debe compilar primero el archivo.l, después el archivo con extensión .yy.c y finalmente el ejecutable como se muestra en la imagen adicionalmente se debe enviar por línea de comando el archivo a analizar.

```
Compiladores]$ flex Programa1.l
Compiladores]$ gcc lex.yy.c -lfl
Compiladores]$ ./a.out Prueba.c
```

La salida serán dos archivos en uno se mostrarán los errores encontrados en el análisis y en el segundo se encontrara de manera general el token generado y su tipo, no se enviaran tablas a ningún archivo.

Conclusiones:
Cuevas Salgado Carlos
Al hacer el analizador léxico tuve mejor entendimiento del lenguaje Lex, además de como poder usarlo con lenguaje C ya que tenia ciertas dudas. La herramienta lex fue de gran ayuda para este programa ya que hizo los tokens y también para poder clasificar ciertos errores léxicos y mostrarlos al usuario. En la parte del llenado de las tablas hubo un problema con el apuntador yytext el cual enviaba información adicional además del token esto por un error en el concepto de apuntador que se resolvía con aun apuntador auxiliar dinámico.

Galindo Chávez Lisset América
Gracias a lo elaborado en este programa logré entender como es que trabaja un analizador léxico, desde que es lo que recibe como entrada hasta que retorna como

salida. Además con el uso de Flex fue más sencillo desarrollar este programa, gracias a las herramientas que posee. Por otro lado, debido a que me toco la parte del desarrollo de tablas junto con su inserción, búsqueda e impresión, me di cuenta que para poder ingresar un texto obtenido a partir de yytext, no se puede hacer directamente ya que copia todo el texto y no únicamente la palabra, para solucionar esto se tuvo que usar una variable auxiliar para luego agregarse al nodo de la tabla correspondiente.