

Práctica Nro. 05

1.- Objetivo.

Escribir analizadores léxicos y ensamblarlos con los analizadores sintácticos predictivos.

2.- Desarrollo.

A) Escribir un analizador sintáctico de infija a postfija incluyendo el analizador léxico.

```
/* Analizador sintáctico de infija a postfija
Exp --> Term Resto
Resto --> mas Term {printf("+")} Resto
Resto --> menos Term {printf("-")} Resto
Resto -->
Term --> {printf(num.valor)} num
*/
```

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#define MAS '+'
#define MENOS '-'
#define NUM 256
#define FIN -1
```

```
char lexema[80];
int tok;

parea(int);
error();
Resto();
Term();
int scanner();
```

```
Exp()
{
    if (tok==NUM)
    { Term();
      Resto(); }
    else
    error();
}
```

```
Resto()
{
    if (tok == MAS)
    { parea(MAS);
      Term();
      printf("+");
      Resto();
    }
    else if (tok == MENOS)
    { parea(MENOS);
      Term();
      printf("-");
      Resto();
    }
    else //cadena vacia
    ;
}
```

```
Term()
{
    if (tok==NUM)
    { printf("%s",lexema);
      parea(NUM); }
    else
    error();
}
```

```
error()
{
    printf("Error de sintaxis");
}
```

```
parea(int t)
{
    if (tok==t)
        tok=scanner();
    else
        error();
}
```

```
int scanner()
{
    int c,i;

    do c=getchar(); while(c==' ');

    if (c=='\n')
        return FIN;

    if (c==MAS || c==MENOS)
        return c;

    if(isdigit(c))
    {
        i=0;
        do { lexema[i++]=c; c=getchar();
        }while(isdigit(c));
        lexema[i]=0;
        ungetc(c,stdin);
        return NUM;
    }
}
```

```
main()
{
    tok=scanner();
    Exp();
}
```

- B) Desarrollar un programa para evaluar expresiones aritméticas por medio de una pila. La entrada se encuentra en su forma postfija. Ejemplo: 52 13 +
- C) Proponga una gramática para la evaluación de expresiones aritméticas. El programa debe leer un archivo de texto plano (programa1.txt), generar la forma postfija y mostrar el resultado.

Programa1.txt

```
INICIO
    14 + 5;
    (12 - 6 ) + 8;
    ((8+12)-(5-2))*2;
FIN
```

3.- Evaluación.

La evaluación consiste en:

- a) (2pts) Solución del ejercicio A en clase de sesión práctica. Deberá enviar el código y la captura de pantalla del programa en ejecución en un archivo PDF.
- b) (2pts) Solución del ejercicio B en clase de sesión práctica. Deberá enviar el código y la captura de pantalla del programa en ejecución en un archivo PDF.
- c) (16pts) Informe del ejercicio C resuelto en un documento en formato PDF. El informe debe contener: enunciado del problema, código fuente y capturas de pantalla del programa en ejecución. De darse el caso, también debe incluir los otros archivos de texto plano

4.- Tiempo de desarrollo.

Tiempo total en sesión práctica: 80 min.