

# ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA DE SISTEMAS CIENCIAS DE LA COMPUTACIÓN**

**COMPILADORES Y LENGUAJES**

# Informe – Prueba 6

**GRUPO:** GR1COM

## FECHA DE ENTREGA: 12/03/2021

**INTEGRANTES:** Alejandro Moya, Leonardo Andrade, Jefferson Cando

Usamos LEX y YACC para implementar una pequeña calculadora que nos permita realizar las operaciones aritméticas con operadores binarios como suma, resta, multiplicación, división y potencia con paréntesis y operandos de tipo entero y real. La calculadora a más de calcular el valor expone la secuencia en notación postfija de manera descrita.

**CÓDIGO LEX**

|  |
| --- |
| %{  #include <stdio.h>  #include "y.tab.h"  FILE \*salida;  %}  %option noyywrap  %option yylineno  DIGITO [0-9]  ID [A-Za-z][a-zA-Z0-9\_]\*  %%  {DIGITO}+("."{DIGITO}+)? {yylval.real=atof(yytext); return(TKN\_NUM);}  "=" {return(TKN\_ASIGN);}  ";" {return(TKN\_PTOCOMA);}  "\*" {return(TKN\_MULT);}  "/" {return(TKN\_DIV);}  "+" {return(TKN\_MAS);}  "-" {return(TKN\_MENOS);}  "^" {return(TKN\_POW);}  "(" {return(TKN\_PARENTESISI);}  ")" {return(TKN\_PARENTESISD);}  {ID} {return(TKN\_ID);}  %%  int main (int argc, char \*argv[]){  if((yyin=fopen("ingreso.txt", "rt"))==NULL)  {  printf("\nNo se puede abrir el archivo: %s\n", argv[1]);  }else  {  if((salida=fopen("salida.txt", "w"))== NULL)  {  printf("\nNo se puede abrir el archivo: %s\n", argv[1]);  }  yyparse();  }  fclose(salida);  fclose(yyin);  return 0;  } |

**CÓDIGO YACC**

|  |
| --- |
| %{  #include <stdio.h>  #include <math.h>  extern int yylex(void);  extern char \*yytext;  FILE \*salida;  void yyerror (char \*s);  %}  %union  {  float real;  }  %start Calculadora  %token <real> TKN\_NUM  %token TKN\_ASIGN  %token TKN\_PTOCOMA  %token TKN\_MULT  %token TKN\_DIV  %token TKN\_MAS  %token TKN\_MENOS  %token TKN\_POW  %token TKN\_PARENTESISI  %token TKN\_PARENTESISD  %token <real> TKN\_ID  %type <real> Expresion  %left TKN\_MAS TKN\_MENOS  %left TKN\_MULT TKN\_DIV  %right TKN\_POW  %%  Calculadora: TKN\_ID {  printf("El resultado de %s con los siguientes TOKENS encontrados es : \nidentificador, asigancion, ",yytext);  fprintf(salida,"El resultado de %s con los siguientes TOKENS encontrados es : \nidentificador, asigancion, ",yytext);  }  TKN\_ASIGN Expresion TKN\_PTOCOMA  {  printf("punto y coma =%5.2f",$4);  fprintf(salida,"punto y coma =%5.2f",$4);  };  Expresion : TKN\_NUM  {  $$=$1;  printf ("numero positivo, ");  fprintf (salida,"numero positivo, ");  }|  TKN\_MENOS TKN\_NUM  {  $$=-$2;  printf("NUmero negativo, ");  fprintf(salida,"NUmero negativo, ");  }|  TKN\_PARENTESISI TKN\_NUM TKN\_PARENTESISD  {  $$=$2;  printf("abre parentesis, numero positivo, cierra parentesis, ");  fprintf(salida,"abre parentesis, numero positivo, cierra parentesis, ");  }|  TKN\_PARENTESISI TKN\_MENOS TKN\_NUM TKN\_PARENTESISD  {  $$=-$3;  printf("abre parentesis, numero negativo, cierra parentesis, ");  fprintf(salida,"abre parentesis, numero negativo, cierra parentesis, ");  }|  Expresion TKN\_POW Expresion  {  $$=pow($1,$3);  printf("operador exponencial, ");  fprintf(salida,"operador exponencial, ");  }|  TKN\_PARENTESISI Expresion TKN\_POW Expresion TKN\_PARENTESISD  {  $$=pow($2,$4);  printf("abre parentesis, operador exponencial, cierra parentesis, ");  fprintf(salida,"abre parentesis, operador exponencial, cierra parentesis, ");  }|  Expresion TKN\_MAS Expresion  {  $$=$1+$3;  printf("operador mas ");  fprintf(salida,"operador mas ");  }|  TKN\_PARENTESISI Expresion TKN\_MAS Expresion TKN\_PARENTESISD  {  $$=$2+$4;  printf("abre parentesis, operador mas, cierra parentesis, ");  fprintf(salida,"abre parentesis, operador mas, cierra parentesis, ");  }|  Expresion TKN\_MENOS Expresion  {  $$=$1-$3;  printf("operador menos ");  fprintf(salida,"operador menos ");  }|  TKN\_PARENTESISI Expresion TKN\_MENOS Expresion TKN\_PARENTESISD  {  $$=$2-$4;  printf("abre parentesis, operador menos, cierra parentesis, ");  fprintf(salida,"abre parentesis, operador menos, cierra parentesis, ");  }|  Expresion TKN\_MULT Expresion  {  $$=$1\*$3;  printf("operador multiplicacion ");  fprintf(salida,"operador multiplicacion ");  }|  TKN\_PARENTESISI Expresion TKN\_MULT Expresion TKN\_PARENTESISD  {  $$=$2\*$4;  printf("abre parentesis, operador multiplicacion, cierra parentesis, ");  fprintf(salida,"abre parentesis, operador multiplicacion, cierra parentesis, ");  }|  Expresion TKN\_DIV Expresion  {  $$=$1/$3;  printf("operador division ");  fprintf(salida,"operador division ");  }|  TKN\_PARENTESISI Expresion TKN\_DIV Expresion TKN\_PARENTESISD  {  $$=$2/$4;  printf("abre parentesis, operador division, cierra parentesis, ");  fprintf(salida,"abre parentesis, operador division, cierra parentesis, ");  };  %%  void yyerror(char \*s){  printf("\nError %s", s);  fprintf(salida,"\nError %s", s);    } |

|  |  |
| --- | --- |
| Entrada | Salida |
| X=1+2+3+4;  X=2+2;  X=2^2^2^2; | El resultado de X con los siguientes TOKENS encontrados es :  identificador, asigancion, numero positivo, numero positivo, operador mas numero positivo, operador mas numero positivo, operador mas punto y coma =10.00  Error syntax error |