EXAMEN PARCIAL 2024-A

Tema: Examen Parcial

Nota

Estudiante	Escuela	Asignatura
Piero Omar De La Cruz Mancilla pdelacruzm@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Compiladores Semestre: V Código: 3.5.6.21

Laboratorio	Tema	Duración
09	Examen Parcial	06 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	29 Abril 2024	13 Mayo 2024

Índice

1. Introducción

Como estudiante apasionado por la programación, me encontré enfrentando constantemente obstáculos en los lenguajes existentes, desde complejidades sintácticas hasta limitaciones en el manejo de datos. Motivado por la necesidad de un enfoque más intuitivo y flexible, decidí crear mi propio lenguaje de programación. Mi objetivo es proporcionar una herramienta accesible para principiantes y lo suficientemente poderosa para satisfacer las demandas de los desarrolladores más experimentados. Con este proyecto, espero fomentar la creatividad y la innovación en la comunidad de programación, ofreciendo una plataforma donde cualquier persona pueda transformar sus ideas en realidad.

2. Especificacion lexica:

- **MAIN**: Marca el inicio del programa principal.
  Expresión regular: `r'MAIN'`
- **FINMAIN**: Marca el final del programa principal.
  Expresión regular: `r'FINMAIN'`

- **DEF**: Marca el inicio de una definición.
  Expresión regular: `r'DEF'`
- **FINDEF**: Marca el final de una definición.
  Expresión regular: `r'FINDEF'`
- **ENT**: Representa el tipo de dato entero.
  Expresión regular: `r'ENT'`
- **FLOT**: Representa el tipo de dato flotante.
  Expresión regular: `r'FLOT'`
- **DOBLE**: Representa el tipo de dato double.
  Expresión regular: `r'DOBLE'`
- **CARAC**: Representa el tipo de dato caracter.
  Expresión regular: `r'CARAC'`
- **BOOL**: Representa el tipo de dato booleano.
  Expresión regular: `r'BOOL'`
- **STRING**: Representa el tipo de dato cadena de caracteres.
  Expresión regular: `r'STRING'`
- **NUM**: Representa un número.
  Expresión regular: `r'\d+(\.\d+)?'`
- **BOOLTR**: Representa el valor booleano "true".
  Expresión regular: `r'true'`
- **BOOLFS**: Representa el valor booleano "false".
  Expresión regular: `r'false'`
- **SI**: Marca el inicio de una estructura condicional "si".
  Expresión regular: `r'SI'`
- **SINO**: Marca el inicio de la parte alternativa de una estructura condicional "si".
  Expresión regular: `r'SINO'`
- **FINSI**: Marca el final de una estructura condicional "si".
  Expresión regular: `r'FINSI'`
- **POR**: Marca el inicio de una estructura de bucle "por".
  Expresión regular: `r'POR'`
- **FINPOR**: Marca el final de una estructura de bucle "por".
  Expresión regular: `r'FINPOR'`
- **MIENTRAS**: Marca el inicio de una estructura de bucle "mientras".
  Expresión regular: `r'MIENTRAS'`
- **FINMIENTRAS**: Marca el final de una estructura de bucle "mientras".
  Expresión regular: `r'EndWhile'`
- **COUT**: Marca la salida de datos en consola.
  Expresión regular: `r'COUT'`
- **CIN**: Marca la entrada de datos desde consola.
  Expresión regular: `r'cin'`

  - **RETORNAR**: Marca el retorno de un valor desde una función.
    Expresión regular: `r'RETORNAR'`
  - **PAR_IZQ**: Representa el paréntesis izquierdo.
    Expresión regular: `r'\('`
  - **PAR_DER**: Representa el paréntesis derecho.
    Expresión regular: `r'\)'`
  - **AND**: Representa el operador lógico .ªND".
    Expresión regular: `r'&&'`
  - **OR**: Representa el operador lógico .ºR".
    Expresión regular: `r'\||'`

- **PYCOMA**: Representa el punto y coma.
  Expresión regular: `r';'`
- **OP_SUMA**: Representa el operador de suma.
  Expresión regular: `r'\+'`
- **OP_RESTA**: Representa el operador de resta.
  Expresión regular: `r'-'`
- **OP_MULTI**: Representa el operador de multiplicación.
  Expresión regular: `r'\*'`
- **OP_DIVI**: Representa el operador de división.
  Expresión regular: `r'/'`
- **OP_IGUAL**: Representa el operador de asignación.
  Expresión regular: `r'='`
- **OP_MAYOR**: Representa el operador de comparación mayor que.
  Expresión regular: `r'>'`
- **OP_MENOR**: Representa el operador de comparación menor que.
  Expresión regular: `r'<'`
- **OP_MAYORIGUAL**: Representa el operador de comparación mayor o igual que.
  Expresión regular: `r'>='`
- **OP_MENORIGUAL**: Representa el operador de comparación menor o igual que.
  Expresión regular: `r'<='`
- **OP_IGUAL_QUE**: Representa el operador de comparación igual que.
  Expresión regular: `r'=='`
- **OP_DIFERENTE_QUE**: Representa el operador de comparación diferente que.
  Expresión regular: `r'!='`
- **CORCH_IZQ**: Representa el corchete izquierdo.
  Expresión regular: `r'\['`
- **CORCH_DER**: Representa el corchete derecho.
  Expresión regular: `r'\]'`
- **COMENTARIOLINEAL**: Representa un comentario de una línea.
  Expresión regular: `r'//.*'`

- **COMILLA**: Representa una cadena de caracteres.
  Expresión regular: `r'\'.*?\''`
- **OP_MOD**: Representa el operador módulo.
  Expresión regular: `r'%'`
- **decimal**: Representa un número decimal.
  Expresión regular: `r'[0-9]+\.[0-9]+'`
- **OP_SUMAUNOAUNO**: Representa el operador de incremento.
  Expresión regular: `r'\+\+'`
- **OP_NEGACION**: Representa el operador de negación lógica.
  Expresión regular: `r'!'`
- **COMA**: Representa la coma.
  Expresión regular: `r','`
- **IMPRIMIR**: Marca la impresión de texto.
  Expresión regular: `r'("[a-zA-Z0-9 ]*")'`
- **ID**: Representa un identificador de variable o función.
  Expresión regular: `r'_[a-zA-Z][a-zA-Z0-9]*_'`

3. Gramatica

```
QQ -> MAIN E FINMAIN
QQ -> DEF ID Parametros E FINDEF QQ
Parametros -> PAR_IZQ VariosPara PAR_DER
VariosPara -> J ID VariosPara2
VariosPara2 -> COMA VariosPara
VariosPara2 -> ''

E -> ASIGNAR
E -> INTERAC
E -> COUT
E -> CIN1
E -> ESTRUCTCONTROL
E -> RETORNAR1
E -> FUNCION E
E -> ''

COUT0 -> COUT COUT1 E
COUT1 -> ID COUT2
COUT1 -> NUM COUT2
COUT1 -> FUNCION COUT2
COUT2 -> OP_SUMA COUT1
COUT2 -> ''

ASIGNAR -> J ID K E
J -> ENT
J -> FLOT
J -> BOOL
J -> STRING
```

```
ELBOOL -> BOOL_TR
ELBOOL -> BOOL_FS

K -> OP_IGUAL O1
K -> ''

O1 -> PAR_IZQ O1 PAR_DER O2
O1 -> ID O2
O1 -> NUM O2
O1 -> FUNCION O2
O1 -> ELBOOL
O1 -> OP_SUMA O1
O2 -> OL O1
O2 -> ''

OL -> OP_SUMA
OL -> OP_RESTA
OL -> OP_MULTI
OL -> OP_DIVI

INTERAC -> ID OP_IGUAL O1 E

ESTRUCTCONTROL -> SII
ESTRUCTCONTROL -> MIENTRAS1
ESTRUCTCONTROL -> POR1

SII -> SI CONDICION E SINOO FINSI E
SINOO -> SINO E
SINOO -> ''
CONDICION -> PAR_IZQ EXPRE PAR_DER
EXPRE -> ID O2 EXPRE1
EXPRE -> NUM O2 EXPRE1
EXPRE -> ELBOOL EXPRE1
EXPRE -> FUNCION O2 EXPRE1
EXPRE1 -> OLC EXPRE
EXPRE1 -> ''

MIENTRAS1 -> MIENTRAS CONDICION E FINMIENTRAS E

POR1 -> POR PAR_IZQ CONDICIONP PAR_DER E FINPOR E
CONDICIONP -> J ID OP_IGUAL NUM COMA EXPRE COMA ID

OLC -> OP_MAYOR
OLC -> OP_MENOR
OLC -> OP_IGUAL_QUE
OLC -> OP_DIFERENTE_QUE
OLC -> OP_IGUAL_QUE
OLC -> OP_MAYORIGUAL
OLC -> OP_MENORIGUAL
OLC -> AND
OLC -> OR
```

```
FUNCION -> ID PAR_IZQ funExp1 PAR_DER
funExp1 -> O1 funExp2
funExp2 -> COMA funExp1
funExp2 -> ''

CIN1 -> CIN ID E

RETORNAR1 -> RETORNAR O1 E
```