

A)

programa -> PROGRAMA VARIABEL bloco FIM

bloco -> INICIO declaracoes comandos

declaracoes -> tipo lista_variaveis | declaracoes tipo lista_variaveis

tipo -> TIPO_INTEIRO | TIPO_REAL | TIPO_CARACTER | TIPO_LISTA_INT |
TIPO_LISTA_REAL

lista_variaveis -> VARIABEL | lista_variaveis VIRGULA VARIABEL | lista_variaveis VIRGULA
CADEIA

comandos -> comando | comandos comando

comando -> atribuicao | escrita | leitura | condicao | repeticao | acesso_lista | CADEIA

acesso_lista -> ABRE_COLCHETE CONSTANTE_INTEIRA FECHA_COLCHETE |
ABRE_COLCHETE VARIABEL FECHA_COLCHETE

condicao -> SE expressao ENTAO comandos FIM_SE

repeticao -> ENQUANTO expressao comandos FIM_ENQUANTO

atribuicao -> VARIABEL ATRIBUICAO expressao

escrita -> ESCREVA ABRE_PARENTESE lista_argumentos FECHA_PARENTESE |
ESCREVA lista_argumentos

lista_argumentos -> expressao | CADEIA | lista_argumentos VIRGULA expressao

leitura -> LEIA ABRE_PARENTESE VARIABEL FECHA_PARENTESE | LEIA
ABRE_COLCHETE VARIABEL FECHA_COLCHETE | LEIA VARIABEL

expressao -> CONSTANTE_INTEIRA | CONSTANTE_REAL | VARIABEL | expressao
PRODUTO expressao | expressao DIVISAO CONSTANTE_INTEIRA | expressao DIVISAO
CONSTANTE_REAL | expressao ADICAO expressao | expressao SUBTRACAO expressao |
expressao IGUAL expressao | expressao MAIOR expressao

B)

Foi desenvolvido um arquivo calcbasica.lex que gera um arquivo tokens.txt contendo todos os tokens gerados durante sua execução.

C), D), E)

Além disso, foi criado um arquivo utilizando a ferramenta Yacc, chamado `calcbasica.y`, onde são realizadas as verificações necessárias para gerar a saída correta.

No arquivo `calcbasica.y`, foram implementadas as operações solicitadas nas questões D) e E). Ele gera uma saída `tabelaSimbolos.txt` com os valores da tabela de símbolos, além de realizar as verificações do analisador semântico.

Também foram desenvolvidos dois arquivos exclusivos para o funcionamento do analisador semântico: `tabela.c` e `tabela.h`. Neles, são armazenadas funções para guardar os dados na tabela e verificar certos tipos de operações.

Como Executar o Código

Para rodar o flex:

- `flex calcbasica.lex`

Para rodar o bison. Existem duas formas que utilizei, sendo elas:

- `bison -Wcounterexamples -d calcbasica.y`
- `bison -d -Wall -v calcbasica.y`

Para rodar tudo e poder executar com sucesso todos os analisadores:

- `gcc calcbasica.tab.c lex.yy.c tabela.c -o calcbasica`

E, por fim, para rodar os exemplos fornecido no arquivo:

- `./calcbasica exemplos/exemplo.txt`
- `./calcbasica exemplos/exemplo2.txt`

Todos os Tokens Aceitos

- "PROGRAMA"
- "SE"
- "ENQUANTO"
- "ENTAO"
- "ESCREVA"
- "FIM_SE"
- "FIM_ENQUANTO"
- "FIM"
- "INICIO"
- "LEIA"
- "CARACTER"
- "INTEIRO"
- "LISTA_INT"
- "LISTA_REAL"
- "REAL"
- "+"
- "!="

- "/"
- "*"
- "-"
- ".l."
- ".M."
- "("
- ")"
- "["
- "]"
- ","
- NUMERO [0-9]+
- CONSTANTE_INTEIRA {NUMERO}
- CONSTANTE_REAL {NUMERO}."{NUMERO}+
- ALFABETO [a-zA-Z]
- VARIABEL {ALFABETO}({ALFABETO}|{NUMERO}|_)*
- CADEIA (\['\'+\'])(\"[^\"]+\")
- COMENTARIOS \{[^\}]*\}
- ESPACOS [\t\n]+

Resultados das Execuções dos Arquivos Exemplos

- Exemplo PROGRAMA fatorial_exemplo:

Este exemplo executa com sucesso, sem encontrar nenhum tipo de erro.

- Exemplo PROGRAMA leitura de lista

Neste exemplo, ocorrem diversos problemas. O principal deles é que a expressão "leitura de lista" está separada por espaços, e meu programa identifica isso como um erro, pois seria necessário utilizar um "_". Além disso, há uma variável sendo usada sem ser declarada.