

ASD com procedimentos recursivos para a linguagem MICRO

Procedure bloco;

```
Procedure idtipo;  
  Begin  
    Repeat  
      If símbolo = ident then simbolo := analex(s) else erro(1);  
      If simbolo = igual then simbolo := analex(s) else erro(2);  
      If simbolo = ident then simbolo := analex(s) else erro(1);  
      If simbolo = pontovirgula then simbolo := analex(s) else erro(3)  
    Until simbolo <> ident  
  End;
```

Procedure idvar;

```
  Begin  
    Repeat  
      If simbolo = ident then simbolo := analex(s) else erro(1);  
      If simbolo = doisPontos then simbolo := analex(s) else erro(4);  
      If símbolo = ident then simbolo := analex(s) else erro(1);  
      If simbolo := pontovirgula then simbolo := analex(s) else erro(3)  
    Until simbolo <> ident  
  End;
```

Procedure decl;

```
  Begin  
    If simbolo = stipo then  
      Begin  
        Simbolo := analex(s); Idtipo  
      End;  
    If símbolo = svar then  
      Begin  
        Simbolo := analex(s); Idvar  
      End  
    End  
  End;
```

Procedure exp;

Procedure termo;

Procedure fator;

```
  Begin  
    Case simbolo of  
      Ident, numero: simbolo := analex(s);  
      abrepar: begin  
        simbolo := analex(s); exp;  
        if símbolo = fecharpar then símbolo := analex(s) else erro(6)  
      end  
      else erro('ident, numero ou ( esperados')  
    end;  
  end;  
  
  begin { termo}  
    fator;
```

```

        while símbolo in [mult, divi] do
            begin
                simbolo := analex(s); fator
            end
        end;

begin {exp}
    if símbolo in [mais, menos] then símbolo := analex(s);
    termo;
    while símbolo in [mais, menos] do
        begin
            simbolo := analex(s); Termo
        end
    end;
end;

```

```

Procedure listaid;
Begin
    Repeat
        Simbolo := analex(s);
        If símbolo = ident then simbolo := analex(s) else erro(1)
    Until simbolo <> virgula
End;

```

```

Procedure listaexp;
Begin
    Repeat
        Simbolo := analex(s); Exp
    Until simbolo <> virgula
End;

```

```

Procedure comandos;
Procedure coms;
Begin
    If símbolo = sread then
        Begin
            Simbolo := analex(s);
            If símbolo <> abrepar then erro(5);
            Listaid;
            if símbolo = fechapar then símbolo := analex(s) Else erro(6)
        End
    Else
    if símbolo = ident then
        Begin
            Simbolo := analex(s);
            If símbolo = atrib then símbolo := analex(s) else erro(7);
            Exp
        End
    Else
    if símbolo = swrite then
        begin
            simbolo := analex(s);

```

```

        if símbolo <> abrepar then erro(5);
        listaexp;
        if simbolo = fecharpar then simbolo := analex(s) else erro(6)
    end
else
if símbolo = sif then
    begin
        simbolo := analex(s) ;
        exp;
        if símbolo = sthen then símbolo := analex(s) else erro(8);
        coms;
        if simbolo = selse then
            begin
                simbolo := analex(s); coms
            end
        end
    end
else
if simbolo = sinicio then
    begin
        simbolo := analex(s) ;
        comandos;
        if símbolo = sfim then analex(s) else erro(9)
    end
else erro('comando esperado')
end; { coms}

```

```

begin { comandos}
coms;
while simbolo = pontovirgula do coms
end;

```

```

begin {bloco}
decl;
if símbolo = sinicio then símbolo := analex(s) else erro(10);
comandos;
if símbolo = sfim then símbolo := analex(s) else erro(9)
end;

```

```

procedure programa;
begin
    bloco;
    if símbolo = ponto then símbolo := analex(s) else erro(11);
end;

```

```

BEGIN {PROGRAMA PRINCIPAL}
Símbolo := analex(s);
Programa;
If símbolo <> eof then erro(12)
END.

```