

Trabalho Prático 2

Partes 1 e 2 - Teoria da Computação e Compiladores SCC605

Grupo:

Lais Pessine do Carmo	7546760
Raphael Victor Ferreira	7143889
Roberto Martins de Freitas	7572940

1. Teste 1

❑ Entrada

```
void main()
{
    typedef int num;
    num a;
    int i;
    int b;
    int aux_var;
    i = 1;
    a = 200;
    b = 0;

    while(i <= a)
    {
        b = i*i + b;
        i = i + 1;

        if (b > a)
        {
            i = 0;
        }
        else
        {
            i = i*i;
        }
    }
}
```

❏ Saída

VOID -> void
IDENT -> main
LPAREN -> (
RPAREN ->)
LBRACE -> {
TYPEDEF -> typedef
IDENT -> int
IDENT -> num
SEMICOLON -> ;
IDENT -> num
IDENT -> a
SEMICOLON -> ;
IDENT -> int
IDENT -> i
SEMICOLON -> ;
IDENT -> int
IDENT -> b
SEMICOLON -> ;
IDENT -> int
IDENT -> aux_var
SEMICOLON -> ;
IDENT -> i
EQUAL -> =
NUMBER -> 1
SEMICOLON -> ;
IDENT -> a
EQUAL -> =
NUMBER -> 200
SEMICOLON -> ;
IDENT -> b
EQUAL -> =
NUMBER -> 0
SEMICOLON -> ;
WHILE -> while
LPAREN -> (
IDENT -> i
LEQ -> <=
IDENT -> a
RPAREN ->)
LBRACE -> {
IDENT -> b

EQUAL -> =
IDENT -> i
MULT -> *
IDENT -> i
PLUS -> +
IDENT -> b
SEMICOLON -> ;
IDENT -> i
EQUAL -> =
IDENT -> i
PLUS -> +
NUMBER -> 1
SEMICOLON -> ;
IF -> if
LPAREN -> (
IDENT -> b
GREATER -> >
IDENT -> a
RPAREN ->)
LBRACE -> {
IDENT -> i
EQUAL -> =
NUMBER -> 0
SEMICOLON -> ;
RBRACE -> }
ELSE -> else
LBRACE -> {
IDENT -> i
EQUAL -> =
IDENT -> i
MULT -> *
IDENT -> i
SEMICOLON -> ;
RBRACE -> }
RBRACE -> }
RBRACE -> }

2. Teste 2

❑ Entrada

```
void main ()
{

    int i;
    int b;
    int a = 2;
    i = 0;
    b = 0;

    while(i <= a)
    {
        b = i*i + b;
        i = i + 1;
    }
}
```

❑ Saída

VOID -> void
IDENT -> main
LPAREN -> (
RPAREN ->)
LBRACE -> {
IDENT -> int
IDENT -> i
SEMICOLON -> ;
IDENT -> int
IDENT -> b
SEMICOLON -> ;
IDENT -> int
IDENT -> a
EQUAL -> =

AnalSin encontrou "=" "=" na linha 6, coluna 15.

Quando na verdade, queríamos ter encontrado um desses:

```
" ,"  
, ,  
" "  
, ,
```

Analizador encontrou erros na análise!

3. Teste 3

❑ Entrada

```
void main()
{
    typedef int num a;
    num b;
    num c;
    a = 1;
    b = 2;

    if(a > b || b > a)
    {
        c = a - b;
    }
    else
    {
        c = b - a;
    }
}
```

❑ Saída

VOID -> void
IDENT -> main
LPAREN -> (
RPAREN ->)
LBRACE -> {
TYPEDEF -> typedef
IDENT -> int
IDENT -> num
IDENT -> a

AnalSin encontrou <IDENT> "a " na linha 3, coluna 25.

Quando na verdade, queríamos ter encontrado:

"," ,

Analizador encontrou erros na análise!

4. Teste 4

❑ Entrada

```
void main()
{
    int a;
    A = 1;

    void addToA(b)
    {
        A = A+b
    }
}
```

❑ Saída

VOID -> void
IDENT -> main
LPAREN -> (
RPAREN ->)
LBRACE -> {
IDENT -> int
IDENT -> a
SEMICOLON -> ;
IDENT -> A
EQUAL -> =
NUMBER -> 1
SEMICOLON -> ;
VOID -> void

AnalSin encontrou "void" "void " na linha 6, coluna 9.

Quando na verdade, queríamos ter encontrado um desses:

```
"}" ,  
"if" ,  
"while" ,  
"typedef" ,
```

<IDENT> ,
<IDENT> ,
"while" ,
"if" ,

Analizador encontrou erros na análise!

5. Teste 5

❑ Entrada

```
void main()  
{  
    int a;  
    a = (3 + 5) / 2;  
    float b;  
    b = (10 - a)a;  
    float aux;  
  
    if (c < b)  
    {  
        aux = b;  
        b = c;  
        c = aux;  
    }  
}
```

❑ Saída

VOID -> void
IDENT -> main
LPAREN -> (
RPAREN ->)
LBRACE -> {
IDENT -> int
IDENT -> a
SEMICOLON -> ;
IDENT -> a
EQUAL -> =
LPAREN -> (

NUMBER -> 3
PLUS -> +
NUMBER -> 5
RPAREN ->)
DIV -> /
NUMBER -> 2
SEMICOLON -> ;
IDENT -> float
IDENT -> b
SEMICOLON -> ;
IDENT -> b
EQUAL -> =
LPAREN -> (
NUMBER -> 10
MINUS -> -
IDENT -> a
RPAREN ->)
IDENT -> a

AnalSin encontrou <IDENT> "a " na linha 6, coluna 21.

Quando na verdade, queríamos ter encontrado um desses:

```
" ,"  
,  
" * "  
,  
" % "  
,  
" / "  
,  
" | | "  
,  
" & & "  
,  
" = = "  
,  
" ! = "  
,  
" < "  
,  
" > "  
,  
" < = "  
,  
" > = "  
,
```

Analizador encontrou erros na análise!

6. Teste 6

❏ Entrada

```
void main()  
{
```



```

int a;
int b;
a = 4;
b = 2;

if (a != b)
{
    c = a*b;
}
else
{
    c = a/b;
}

```

❏ Saída

VOID -> void
 IDENT -> main
 LPAREN -> (
 RPAREN ->)
 LBRACE -> {
 IDENT -> int
 IDENT -> a
 SEMICOLON -> ;
 IDENT -> int
 IDENT -> b
 SEMICOLON -> ;
 IDENT -> a
 EQUAL -> =
 NUMBER -> 4
 SEMICOLON -> ;
 IDENT -> b
 EQUAL -> =
 NUMBER -> 2
 SEMICOLON -> ;
 IF -> if
 LPAREN -> (
 IDENT -> a
 DIF -> !=
 IDENT -> b
 RPAREN ->)

```

LBRACE -> {
IDENT -> c
EQUAL -> =
IDENT -> a
MULT -> *
IDENT -> b
SEMICOLON -> ;
RBRACE -> }
ELSE -> else
LBRACE -> {
IDENT -> c
EQUAL -> =
IDENT -> a
DIV -> /
IDENT -> b
SEMICOLON -> ;
RBRACE -> }

```

AnalSin encontrou <EOF> na linha 16, coluna 1.

Quando na verdade, queríamos ter encontrado um desses:

```

"}" ,
"if" ,
"while" ,
"typedef" ,
<IDENT> ,
<IDENT> ,
"while" ,
"if" ,

```

Analizador encontrou erros na análise!

7. Teste 7

❏ Entrada

```

void main()
{
    int x;
    int i;
    x = 1;

```

```

    while ()
    {
        x= x*i;
    }
}

```

❏ Saída

```

VOID -> void
IDENT -> main
LPAREN -> (
RPAREN -> )
LBRACE -> {
IDENT -> int
IDENT -> x
SEMICOLON -> ;
IDENT -> int
IDENT -> i
SEMICOLON -> ;
IDENT -> x
EQUAL -> =
NUMBER -> 1
SEMICOLON -> ;
WHILE -> while
LPAREN -> (
RPAREN -> )

```

AnalSin encontrou ")" " " na linha 7, coluna 16.

Quando na verdade, queríamos ter encontrado um desses:

```

<NUMBER> ,
"(" ,
"+" ,
"_" ,
"!" ,
<IDENT> ,

```

Analizador encontrou erros na análise!

Observação: A descrição dos testes estão no arquivo README no arquivo do trabalho.