

Project 1

Implementation of a Lexer

Due Friday, March 29, 2019

1. Identifiers (10%)

Input:

ABCD OR_AND _EFG AND_1 NOT1 _123 _A1

Output:

(token ABCD OR_AND _EFG AND_1 NOT1 _123 _A1)

2. Keywords (15%)

Input:

begin declare else end exit for if in integer is loop procedure read then
write

Output:

(token begin declare else end exit for if in integer is loop procedure
read then write)

3. Integer constants (10%)

Input:

123 00123 0 4 8 4560

Output:

(token 123 0 0 123 0 4 8 4560)

4. Operators (20%)

Input:

: .. ; + - * / % = < > >= < <= && || ! := ()

Output:

(token : .. ; + - * / % = < > >= < <= && || ! := ())

5. Comments (10%)

Input:

```
// nothing should be printed
```

Output:

token

6. White spaces (10%)

Input:

```
// The following line contains a ' ', a '\t', and a '\n'.
```

Output:

token

7. Example (25%)

Input:

```
// A program to sum 1 to N
```

```
procedure SUM is
```

```
  declare
```

```
    N: integer;
```

```
    S: integer;
```

```
    I: integer;
```

```
begin
```

```
  read N;
```

```
  if N < 0 then
```

```
    write -1;
```

```
    exit;
```

```
  else
```

```
    S := 0;
```

```
end if;

// Loop
for I in 1..N loop
    S := S + I;
end loop;
write S;
end;
```

Output:

(token procedure SUM is declare N : integer ; S : integer ; I : integer ;
begin read N ; if N < 0 then write - 1 ; exit ; else S := 0 ; end if ; for I
in 1 .. N loop S := S + I ; end loop ; write S ; end ;)