**Test Plan for Assignment #2**

Standard Tests with current directory being the debug folder containing project files.

Test 1) The file a2error.out is produced by the following command:

**scanner.exe a2error.pls > a2error.out**

|  |
| --- |
| **Reading file a2error.pls ....Please wait**  **Printing buffer parameters:**  **The capacity of the buffer is: 1036**  **The current size of the buffer is: 1036**  **Printing buffer contents:**  **!!This file contains many lexical errors.**  **!!It tests mainly the correctness of the accepting functions**  **!!5 tabs**        **a1234567**  **b123456#**  **!!legal, but too long (possible semantic error)**  **a1234567MORE**  **b1234567MORE#**  **!!lexical error: # out of context, allowed only in comments and strings**  **#OR@%**  **!!lexical error: | allowed only in strings and comments**  **|**  **!!another lexical error: $ out of context**  **Not$here**  **!!another lexical error: . out of context**  **.**  **!!another lexical error: & illegal symbol**  **&**  **!!legal FPL (will be rounded to 2.0 when displayed)**  **1.999999999999999911111111111111111111111111111111111111111111111111111111111111111**  **!! big float number (error: out of range)**  **999999999999999999999999999999999999999999999999999999999999999999999999999999999.0**  **!!small float number (error: out of range)**  **0.00000000000000000000000000000000000000000000000000000000000000000000000000000000001**  **!! big decimal int number (error: out of range)**  **99999**  **!! gigantic decimal int number (error: out of range)**  **123456789012345678901234567890**    **Scanning source file...**  **Token Attribute**  **----------------------------------**  **AVID\_T a1234567**  **SVID\_T b123456#**  **AVID\_T a1234567**  **SVID\_T b123456#**  **ERR\_T #**  **AVID\_T OR**  **ERR\_T @**  **ERR\_T %**  **ERR\_T |**  **AVID\_T Not**  **ERR\_T $**  **AVID\_T here**  **ERR\_T .**  **ERR\_T &**  **FPL\_T 2.000000**  **ERR\_T 99999999999999999...**  **ERR\_T 0.000000000000000...**  **ERR\_T 99999**  **ERR\_T 12345678901234567...**  **SEOF\_T 0** |

Test 2) The file a2empty.out is produced by the following command:

**scanner.exe a2empty.pls > a2empty.out**

|  |
| --- |
| Reading file a2empty.pls ....Please wait  Printing buffer parameters:  The capacity of the buffer is: 1  The current size of the buffer is: 1  Printing buffer contents:    Scanning source file...  Token Attribute  ----------------------------------  SEOF\_T 0 |

Test 3) The file ass2r.out is produced by the following command:

**scanner.exe ass2r.pls > ass2r.out**

|  |
| --- |
| **Reading file ass2r.pls ....Please wait**  **Printing buffer parameters:**  **The capacity of the buffer is: 582**  **The current size of the buffer is: 582**  **Printing buffer contents:**  **!! This program calculates the sum of 32767 even numbers.**  **!! The program is "lexically" correct**  **!! and should not generate any error**  **PLATYPUS {**    **a=+0.0;**    **sum008 = 7.87050 ;**  **READ(a,sum008);**  **i=0;**  **WHILE TRUE (i < 32767 .OR. i == 32767)REPEAT{**  **i = i + 2;**  **a=**  **a\*i/0.5**  **;**  **sum008 = sum008 + a - 1 ;**  **};**  **IF TRUE(text# == "")THEN {**  **text# = "prog" ## "ram";**  **}**  **ELSE {**  **text# = text# ## "ram";**  **};**  **WRITE("\\* This is a platypus -:)-<-<-- \\*");**  **WRITE(text#);**    **IF FALSE (text# == "program".OR.sum008<>8..AND.i>-10)THEN {**  **WRITE(sum008);**  **WRITE();**  **}**  **ELSE{};**  **}**    **Scanning source file...**  **Token Attribute**  **----------------------------------**  **KW\_T PLATYPUS**  **LBR\_T**  **AVID\_T a**  **ASS\_OP\_T**  **ART\_OP\_T 0**  **FPL\_T 0.000000**  **EOS\_T**  **AVID\_T sum008**  **ASS\_OP\_T**  **FPL\_T 7.870500**  **EOS\_T**  **KW\_T READ**  **LPR\_T**  **AVID\_T a**  **COM\_T**  **AVID\_T sum008**  **RPR\_T**  **EOS\_T**  **AVID\_T i**  **ASS\_OP\_T**  **INL\_T 0**  **EOS\_T**  **KW\_T WHILE**  **KW\_T TRUE**  **LPR\_T**  **AVID\_T i**  **REL\_OP\_T 3**  **INL\_T 32767**  **LOG\_OP\_T 1**  **AVID\_T i**  **REL\_OP\_T 0**  **INL\_T 32767**  **RPR\_T**  **KW\_T REPEAT**  **LBR\_T**  **AVID\_T i**  **ASS\_OP\_T**  **AVID\_T i**  **ART\_OP\_T 0**  **INL\_T 2**  **EOS\_T**  **AVID\_T a**  **ASS\_OP\_T**  **AVID\_T a**  **ART\_OP\_T 2**  **AVID\_T i**  **ART\_OP\_T 3**  **FPL\_T 0.500000**  **EOS\_T**  **AVID\_T sum008**  **ASS\_OP\_T**  **AVID\_T sum008**  **ART\_OP\_T 0**  **AVID\_T a**  **ART\_OP\_T 1**  **INL\_T 1**  **EOS\_T**  **RBR\_T**  **EOS\_T**  **KW\_T IF**  **KW\_T TRUE**  **LPR\_T**  **SVID\_T text#**  **REL\_OP\_T 0**  **STR\_T 0**  **RPR\_T**  **KW\_T THEN**  **LBR\_T**  **SVID\_T text#**  **ASS\_OP\_T**  **STR\_T 1 prog**  **SCC\_OP\_T**  **STR\_T 6 ram**  **EOS\_T**  **RBR\_T**  **KW\_T ELSE**  **LBR\_T**  **SVID\_T text#**  **ASS\_OP\_T**  **SVID\_T text#**  **SCC\_OP\_T**  **STR\_T 10 ram**  **EOS\_T**  **RBR\_T**  **EOS\_T**  **KW\_T WRITE**  **LPR\_T**  **STR\_T 14 \\* This is a platypus -:)-<-<-- \\***  **RPR\_T**  **EOS\_T**  **KW\_T WRITE**  **LPR\_T**  **SVID\_T text#**  **RPR\_T**  **EOS\_T**  **KW\_T IF**  **KW\_T FALSE**  **LPR\_T**  **SVID\_T text#**  **REL\_OP\_T 0**  **STR\_T 49 program**  **LOG\_OP\_T 1**  **AVID\_T sum008**  **REL\_OP\_T 1**  **FPL\_T 8.000000**  **LOG\_OP\_T 0**  **AVID\_T i**  **REL\_OP\_T 2**  **ART\_OP\_T 1**  **INL\_T 10**  **RPR\_T**  **KW\_T THEN**  **LBR\_T**  **KW\_T WRITE**  **LPR\_T**  **AVID\_T sum008**  **RPR\_T**  **EOS\_T**  **KW\_T WRITE**  **LPR\_T**  **RPR\_T**  **EOS\_T**  **RBR\_T**  **KW\_T ELSE**  **LBR\_T**  **RBR\_T**  **EOS\_T**  **RBR\_T**  **SEOF\_T 0**  **prog ram ram \\* This is a platypus -:)-<-<-- \\* program** |

Test 4) The file ass2w.out is produced by the following command:

**scanner.exe ass2w.pls > ass2w.out**

|  |
| --- |
| **Reading file ass2w.pls ....Please wait**  **Printing buffer parameters:**  **The capacity of the buffer is: 971**  **The current size of the buffer is: 971**  **Printing buffer contents:**  **!!This program contains many lexical errors**  **!!It tests mainly your transition table**  **!! Use MY output to adjust**  **!! YOUR error state transitions in YOUR transition table**  **!!You should create your own test file with more errors**  **!1Wrong comment**  **PLATYPUs {**  **i=000; !! legal zero decimal literal**  **n=001; !! illegal decimal literal**  **j=0#; !! syntax error**  **k=2I; !! syntax error**  **k=0O; !! digit 0 followed by a letter O**  **a% = 0.0L; !! syntax error**  **b. = 0.L !! syntax error**  **USING> = .0 !! lexical errors,illegal floating point literal**  **Read(ElSe)**  **If=0.l !! letter l not digit 1**  **o=0x1Ga !! syntax error**  **b10 = 1O1.0; !! letter O follows 1**  **3C=7.0O2; !! digit in VID and letter O precedes 2**  **name#S = Last# ## " S.#" !! wrong string VID**  **WHILE FALSE(b10 > 0x00 OR. <= b10.AND a != o.o)LOOP{**  **a=a+1..;**  **c=001.1; !! leading zero not allowed in floating point**  **}**  **WRITE##("No Luck Today);**  **WRITE (""My Scanner has gone astray");**  **} ! !Wrong comment**  **"There is always one more bug**  **Scanning source file...**  **Token Attribute**  **----------------------------------**  **ERR\_T !1**  **AVID\_T PLATYPUs**  **LBR\_T**  **AVID\_T i**  **ASS\_OP\_T**  **INL\_T 0**  **EOS\_T**  **AVID\_T n**  **ASS\_OP\_T**  **ERR\_T 001**  **EOS\_T**  **AVID\_T j**  **ASS\_OP\_T**  **INL\_T 0**  **ERR\_T #**  **EOS\_T**  **AVID\_T k**  **ASS\_OP\_T**  **ERR\_T 2I**  **EOS\_T**  **AVID\_T k**  **ASS\_OP\_T**  **ERR\_T 0O**  **EOS\_T**  **AVID\_T a**  **ERR\_T %**  **ASS\_OP\_T**  **FPL\_T 0.000000**  **AVID\_T L**  **EOS\_T**  **AVID\_T b**  **ERR\_T .**  **ASS\_OP\_T**  **FPL\_T 0.000000**  **AVID\_T L**  **AVID\_T USING**  **REL\_OP\_T 2**  **ASS\_OP\_T**  **ERR\_T .**  **INL\_T 0**  **AVID\_T Read**  **LPR\_T**  **AVID\_T ElSe**  **RPR\_T**  **AVID\_T If**  **ASS\_OP\_T**  **FPL\_T 0.000000**  **AVID\_T l**  **AVID\_T o**  **ASS\_OP\_T**  **ERR\_T 0x**  **ERR\_T 1G**  **AVID\_T a**  **AVID\_T b10**  **ASS\_OP\_T**  **ERR\_T 1O**  **FPL\_T 1.000000**  **EOS\_T**  **ERR\_T 3C**  **ASS\_OP\_T**  **FPL\_T 7.000000**  **AVID\_T O2**  **EOS\_T**  **SVID\_T name#**  **AVID\_T S**  **ASS\_OP\_T**  **SVID\_T Last#**  **SCC\_OP\_T**  **STR\_T 0 S.#**  **KW\_T WHILE**  **KW\_T FALSE**  **LPR\_T**  **AVID\_T b10**  **REL\_OP\_T 2**  **ERR\_T 0x**  **INL\_T 0**  **AVID\_T OR**  **ERR\_T .**  **REL\_OP\_T 3**  **ASS\_OP\_T**  **AVID\_T b10**  **ERR\_T .**  **AVID\_T AND**  **AVID\_T a**  **ERR\_T !=**  **AVID\_T a**  **ASS\_OP\_T**  **AVID\_T a**  **ART\_OP\_T 0**  **FPL\_T 1.000000**  **ERR\_T .**  **EOS\_T**  **AVID\_T c**  **ASS\_OP\_T**  **ERR\_T 001**  **ERR\_T .**  **INL\_T 1**  **EOS\_T**  **RBR\_T**  **SVID\_T WRITE#**  **ERR\_T #**  **LPR\_T**  **STR\_T 5 No Luck Today);**  **WRITE (**  **STR\_T 31 My Scanner has gone astray**  **RPR\_T**  **EOS\_T**  **RBR\_T**  **ERR\_T !**  **ERR\_T "There is always ...**  **SEOF\_T 0**  **S.# No Luck Today);**  **WRITE ( My Scanner has gone astray** |