测试结果分析

测试1（错误）：

在第一次测试过程中词法分析程序报错在分析 “ ‘”（quotation）符号，这因为在源代码本人忘了加一个条件语句让程序识别quotation符号。结果不正确。

测试2（正确）：

在此测试程序已经有改进了，程序会分析quotation符号，但这次测试主要部分是程序能否识别quotation符号，结果正确。

测试3（错误）：

在此测试，重要部分的程序能否识别语句，在这测试中程序识别语句当成IDEN类。结果错误，这因为在代码里，本人没有加识别语句的条件。

测试4（正确）：

在此测试，重要是程序分析单词的值，在这测试程序能分析单词的值，结果正确。

测试5（正确）：

在此测试，重要测试程序能否输出单词的类，这测试中程序能满足要求，程序能输出正确的单词类别，结果正确。

测试6（正确）：

在此测试，重要测试程序能否输出单词的类别编码，这测试中程序能满足要求，程序能输出正确的单词类别码，结果正确。编码是本人自定的。

测试7（正确）：

在此测试，重要测试程序能否输出单词line而能报错在哪有错误，这测试中程序能满足要求，结果正确。

下面是程序的测试分析结果：

Result of the analyse :

line Code Category Value

1 10: IDEN program

1 10: IDEN project1

1 26: SEMICOLON ;

3 32: TKN var

3 10: IDEN x

3 29: COMMA ,

3 10: IDEN y

3 29: COMMA ,

3 10: IDEN m

3 17: COLON :

3 10: IDEN integer

3 26: SEMICOLON ;

4 32: TKN var

4 10: IDEN z

4 17: COLON :

4 10: IDEN double

4 26: SEMICOLON ;

5 31: TKN const

5 10: IDEN a

5 25: SYMBOL =

5 11: NUM 2

5 29: COMMA ,

5 10: IDEN b

5 25: SYMBOL =

5 11: NUM 5

5 29: COMMA ,

5 10: IDEN c

5 25: SYMBOL =

5 11: NUM 3

5 26: SEMICOLON ;

7 33: TKN procedure

7 10: IDEN message

7 26: SEMICOLON ;

8 1: TKN begin

9 10: IDEN write

9 27: LPAREN (

9 31: QUOTATION '

9 10: IDEN this

9 10: IDEN is

9 10: IDEN test

9 31: QUOTATION '

9 28: RPAREN )

9 26: SEMICOLON ;

10 6: TKN end

10 26: SEMICOLON ;

11 1: TKN begin

12 10: IDEN y

12 18: BECOMES :=

12 11: NUM 1

12 26: SEMICOLON ;

13 10: IDEN read

13 27: LPAREN (

13 10: IDEN x

13 28: RPAREN )

13 26: SEMICOLON ;

14 2: TKN if

14 10: IDEN x

14 20: GTR >

14 10: IDEN y

15 3: TKN then

16 1: TKN begin

17 34: TKN call

17 10: IDEN message

17 26: SEMICOLON ;

18 10: IDEN y

18 18: BECOMES :=

18 10: IDEN y

18 13: MULTI \*

18 11: NUM 2

18 26: SEMICOLON ;

19 10: IDEN writeln

19 27: LPAREN (

19 10: IDEN y

19 28: RPAREN )

19 26: SEMICOLON ;

20 6: TKN end

20 26: SEMICOLON ;

Error in line 21!

21 10: IDEN x

21 18: BECOMES :=

21 10: IDEN x

21 16: MINUS -

21 10: IDEN a

21 26: SEMICOLON ;

Error in line 21!

22 2: TKN if

22 35: TKN odd

22 10: IDEN x

23 3: TKN then

24 1: TKN begin

25 34: TKN call

25 10: IDEN message

25 26: SEMICOLON ;

26 10: IDEN z

26 18: BECOMES :=

26 10: IDEN x

26 13: MULTI \*

26 27: LPAREN (

26 11: NUM 6

26 14: DIVIDE /

26 11: NUM 2

26 28: RPAREN )

26 26: SEMICOLON ;

27 10: IDEN write

27 27: LPAREN (

27 10: IDEN z

27 28: RPAREN )

27 26: SEMICOLON ;

28 6: TKN end

28 26: SEMICOLON ;

29 4: TKN while

29 10: IDEN x

29 20: GTR >

29 11: NUM 0

29 5: TKN do

30 10: IDEN x

30 18: BECOMES :=

30 10: IDEN x

30 16: MINUS -

30 11: NUM 1

30 26: SEMICOLON ;

32 10: IDEN write

32 27: LPAREN (

32 31: QUOTATION '

32 10: IDEN X

32 10: IDEN last

32 10: IDEN value

32 31: QUOTATION '

32 28: RPAREN )

32 26: SEMICOLON ;

33 10: IDEN write

33 27: LPAREN (

33 10: IDEN x

33 28: RPAREN )

33 26: SEMICOLON ;

34 6: TKN end

34 0: PERIOD .