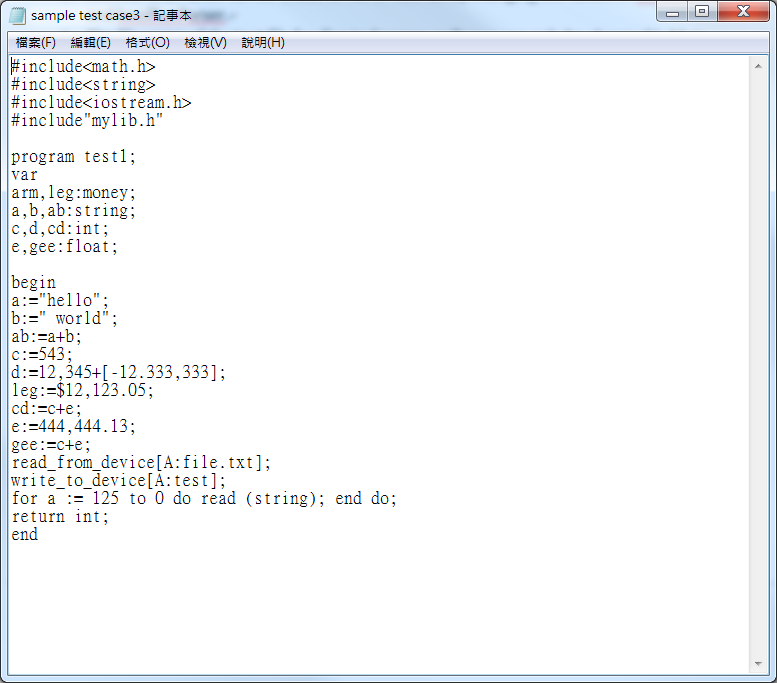
# First set and follow set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | symbo | derive  λ array | first set | follow set |
| 1 | program | f | program | NA |
| 2 | libraraytoken | f | libraraytoken | NA |
| 3 | var | f | var | NA |
| 4 | id | f | id | NA |
| 5 | begin | f | begin | NA |
| 6 | . | f | . | NA |
| 7 | := | f | := | NA |
| 8 | ; | f | ; | NA |
| 9 | , | f | , | NA |
| 10 | : | f | : | NA |
| 11 | ( | f | ( | NA |
| 12 | ) | f | ) | NA |
| 13 | < | f | < | NA |
| 14 | > | f | > | NA |
| 15 | <= | f | <= | NA |
| 16 | >= | f | >= | NA |
| 17 | = = | f | = = | NA |
| 18 | != | f | != | NA |
| 19 | + | f | + | NA |
| 20 | - | f | - | NA |
| 21 | \* | f | \* | NA |
| 22 | / | f | / | NA |
| 23 | int | f | int | NA |
| 24 | real | f | real | NA |
| 25 | scientific | f | scientific | NA |
| 26 | currencylist | f | currencylist | NA |
| 27 | stringliteral | f | stringliteral | NA |
| 28 | abs | f | abs | NA |
| 29 | end | f | end | NA |
| 30 | return | f | return | NA |
| 31 | until | f | until | NA |
| 32 | for | f | for | NA |
| 33 | repeat | f | repeat | NA |
| 34 | while | f | while | NA |
| 35 | if | f | if | NA |
| 36 | do | f | do | NA |
| 37 | then | f | then | NA |
| 38 | println | f | println | NA |
| 39 | else | f | else | NA |
| 40 | read | f | read | NA |
| 42 | write\_to \_device | f | write\_to \_device | NA |
| 43 | device\_open | f | device\_open | NA |
| 44 | device\_close | f | device\_close | NA |
| 45 | read\_from\_device | f | read\_from\_device | NA |
| 46 | $ | f | $ | NA |
| 1 | <Goal> | f | libraraytoken | program| λ | $ |
| 2 | <Libtoken\_Tail> | t | libraraytoken| λ | $| program |
| 3 | <Libtiken> | f | libraraytoken | $|program| libraraytoken |
| 4 | <start> | f | program | $ |
| 5 | <Code> | f | program | $ |
| 6 | <variables> | f | var | begin |
| 7 | <variable\_type\_tail> | t | id | λ | begin |
| 8 | <variable\_type> | f | id | : |
| 9 | <varable\_tail> | t | , | λ | : |
| 10 | <variable> | f | id | :| , |
| 11 | <Main> | f | begin | end |
| 12 | <statement\_list> | f | id |device\_open|for| repeat| while | if|device\_close |read\_from\_device| write\_to\_device |println | read | end | return| until |
| 13 | <statement\_tail> | t | id |device\_open|for| repeat| while | if|device\_close |read\_from\_device| write\_to\_device |println | read | λ | end | return | until |
| 14 | <statement> | f | id |device\_open|for| repeat| while | if|device\_close |read\_from\_device| write\_to\_device |println | read | end | return| until id |device\_open|for| repeat| while | if|device\_close |read\_from\_device| write\_to\_device |println | read |
| 15 | <Booleancondition> | f | \* | / | stringliteral | ) |
| 16 | <relation\_op> | f | < | >| <= | >= | == | != | (|id| int |real | scientific | currenylit | abs | stringliteral |
| 17 | <expressionGversion2> | f | \* | / | stringliteral( | id |int | real | scientific | currencylit | abs | < | >| <= | >= | == | != | ) |
| 18 | <primary\_tail> | t | λ| +| - | < | >| <= | >= | == | != | )|; |
| 19 | <primary> | f | \* | /( | id |int | real | scientific | currencylit | abs | < | >| <= | >= | == | !=| +|- | ) |; |
| 20 | <secondary\_tail> | t | λ| \*| / | < | >| <= | >= | == | !=| +|-| ) |\* | / |
| 21 | <secondary> | f | ( | id |int | real | scientific | currencylit | abs | < | >| <= | >= | == |  !=| +|-| ) |\* | /( | id |int | real | scientific | currencylit | abs |
| 22 | <end> | f | end. | $ |

# Predict set:

|  |  |  |
| --- | --- | --- |
|  | **Production** | **Predict Sets** |
| 1 | <Goal> --> <Libtoken> <Libtoken\_\_Tail><Start> | libraraytoken | program |
| 2 | <Goal>--> <Start> | program |
| 3 | <Libtoken\_\_Tail> --> <Libtoken><Libtoken\_Tail> | libraraytoken |
| 4 | <Libtoken\_\_Tail> --> λ | program |
| 5 | <Libtoken> --> **libraraytoken** | libraraytoken |
| 6 | <Start> --> <Code> | program |
| 7 | <Code> --> **program id**; <variables> < Main > <end> | program |
| 8 | <variables> --> **var** <variable\_type> : **id** ; <variable\_type\_tail> | var |
| 9 | <variable\_\_type\_tail> --> <variable\_type> : **id** ; <variable\_type\_tail> | id |
| 10 | <variable\_\_type\_tail> --> λ | begin |
| 11 | <variable\_type> --> <variable> <variable tail> | id |
| 12 | <variable\_tail> --> , <variable> <variable\_tail> | , |
| 13 | <variable\_tail> --> λ | , | id| begin |
| 14 | <variable> --> **id** | id |
| 15 | <Main> --> **begin** <statement\_list> **return int**; | begin |
| 16 | <statement\_list> --> <statement> <statement tail> | id |device\_open|for| repeat| while | if|device\_close |read\_from\_device| write\_to\_device |println | read |
| 17 | <statemen\_ tail〉 --> <statement> <statement\_tail> | id |device\_open|for| repeat| while | if|device\_close |read\_from\_device| write\_to\_device |println | read |
| 18 | <statemen\_ tail〉 --> λ | end | return | until |
| 19 | <statement> --> **id** := <expressionGversion2>; | id |
| 20 | <statement> --> **device\_open<**filetoken>; | device\_open |
| 21 | <statement> -->**for id := id to id do** <statement\_\_list> **end do**; | for |
| 22 | <statement> -->**repeat do** <statement\_list> **until** (<Booleancondition>) **end do**; | repeat |
| 23 | <statement> --> **while** (<Booleancondition>) **do** <statement\_list> **end do**; | while |
| 24 | <statement> -->**if** (<Booleancondition>) **then begin** <statement\_list> **end** ; **else begin** <staten:ent\_list> end; | If |
| 25 | <statement> **--> device\_close<**filetoken**>;** | device\_close |
| 26 | <statement> --> **read\_from\_device<**filetoken**>**; | read\_from\_device |
| 27 | <statement> -->**write\_to\_device<filetoken>**; | write\_to\_device |
| 28 | <statement> -->**println**(<expressicnGversion2>) ; | println |
| 29 | <statement> --> **read**(<variable type>); | read |
| 30 | <Booleancondition> --> <expressionGversion2> <relational\_op> <expressionGversion2> | \* | / | stringliteral | id |int | real | scientific | currencylit | abs |
| 31 | <relational\_op> --> < | < |
| 32 | <relational\_op> --> > | > |
| 33 | <relational\_op> --> <= | <= |
| 34 | <relational\_op> --> >= | >= |
| 35 | <relational\_op> --> == | == |
| 36 | <relational\_op> --> != | != |
| 37 | <expressionGversion2> --> <primary> <primary\_tail> | \* | /| id |int | real | scientific | currencylit | abs |
| 38 | <expressionGversion2> --> **stringliteral** | sttringliteral |
| 39 | <primary\_tail> --> + <primary><pnmary\_tail> | + |
| 40 | <primary\_tail> --> - <primary> <primary\_\_tail> | - |
| 41 | <primary\_tail> --> λ | < | >| <= | >= | == | !=| )|; |
| 42 | <primary> --> < secondary> <secondary\_tail> | ( | id |int | real | scientific | currencylit | abs |
| 43 | <secondary\_tail> --> \* <secondary><secondary\_tail> | \* |
| 44 | <secondary\_tail> --> / <secondary> <secondary\_tail> | / |
| 45 | <secondary\_tail> --> λ | < | >| <= | >= | == |  !=| +|-| ) |\* | / |
| 46 | <secondary> --> (<expressionGversion2>) | ( |
| 47 | <secondary> --> **id** | id |
| 48 | <secondary> --> **int** | int |
| 49 | <secondary> --> **real** | real |
| 50 | <secondary> --> **scientific** | scientific |
| 51 | <secondary> --> **currencylit** | currencylit |
| 52 | <secondary> --> **abs** (<expressionGversion2>) | abs |
| 54 | <end> -> end. | end. |

# Input Sample\_Test\_Case.txt:



# ScannerOutput.txt:

#include<math.h>

#include<string>

#include<iostream.h>

#include"mylib.h"

program

test1

;

var

arm

,

leg

:

money

;

a

,

b

,

ab

:

string

;

c

,

d

,

cd

:

int

;

e

,

gee

:

float

;

begin

a

:=

"hello"

;

b

:=

" world"

;

ab

:=

a

+

b

;

c

:=

543

;

d

:=

12,345

+

[-12.333,333]

;

leg

:=

$12,123.05

;

cd

:=

c

+

e

;

e

:=

444,444.13

;

gee

:=

c

+

e

;

read\_from\_device

[A:file.txt]

;

write\_to\_device

[A:test]

;

for

a

:=

125

to

0

do

read

(

string

)

;

end

do

;

read

(

string

)

;

end

# Result of Sample\_test\_case:

-----------------------Parser-----------------

Fire 1

Fire 5

Match and pop #include<math.h>

Fire 3

Fire 5

Match and pop #include<string>

Fire 3

Fire 5

Match and pop #include<iostream.h>

Fire 3

Fire 5

Match and pop #include"mylib.h"

Fire 4

Fire 6

Fire 7

Match and pop program

Match and pop test1

Match and pop ;

Fire 8

Match and pop var

Fire 11

Fire 14

Match and pop arm

Fire 12

Match and pop ,

Fire 14

Match and pop leg

Fire 13

Match and pop :

Match and pop money

Match and pop ;

Fire 9

Fire 11

Fire 14

Match and pop a

Fire 12

Match and pop ,

Fire 14

Match and pop b

Fire 12

Match and pop ,

Fire 14

Match and pop ab

Fire 13

Match and pop :

Match and pop string

Match and pop ;

Fire 9

Fire 11

Fire 14

Match and pop c

Fire 12

Match and pop ,

Fire 14

Match and pop d

Fire 12

Match and pop ,

Fire 14

Match and pop cd

Fire 13

Match and pop :

Match and pop int

Match and pop ;

Fire 9

Fire 11

Fire 14

Match and pop e

Fire 12

Match and pop ,

Fire 14

Match and pop gee

Fire 13

Match and pop :

Match and pop float

Match and pop ;

Fire 10

Fire 15

Match and pop begin

Fire 16

Fire 19

Match and pop a

Match and pop :=

Fire 38

Match and pop "hello"

Match and pop ;

Fire 17

Fire 19

Match and pop b

Match and pop :=

Fire 38

Match and pop " world"

Match and pop ;

Fire 17

Fire 19

Match and pop ab

Match and pop :=

Fire 37

Fire 42

Fire 47

Match and pop a

Fire 45

Fire 39

Match and pop +

Fire 42

Fire 47

Match and pop b

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 19

Match and pop c

Match and pop :=

Fire 37

Fire 42

Fire 48

Match and pop 543

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 19

Match and pop d

Match and pop :=

Fire 37

Fire 42

Fire 48

Match and pop 12,345

Fire 45

Fire 39

Match and pop +

Fire 42

Fire 49

Match and pop [-12.333,333]

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 19

Match and pop leg

Match and pop :=

Fire 37

Fire 42

Fire 51

Match and pop $12,123.05

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 19

Match and pop cd

Match and pop :=

Fire 37

Fire 42

Fire 47

Match and pop c

Fire 45

Fire 39

Match and pop +

Fire 42

Fire 47

Match and pop e

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 19

Match and pop e

Match and pop :=

Fire 37

Fire 42

Fire 49

Match and pop 444,444.13

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 19

Match and pop gee

Match and pop :=

Fire 37

Fire 42

Fire 47

Match and pop c

Fire 45

Fire 39

Match and pop +

Fire 42

Fire 47

Match and pop e

Fire 45

Fire 41

Match and pop ;

Fire 17

Fire 26

Match and pop read\_from\_device

Match and pop [A:file.txt]

Match and pop ;

Fire 17

Fire 27

Match and pop write\_to\_device

Match and pop [A:test]

Match and pop ;

Fire 17

Fire 21

Match and pop for

Match and pop a

Match and pop :=

Match and pop 125

Match and pop to

Match and pop 0

Match and pop do

Fire 16

Fire 29

Match and pop read

Match and pop (

Fire 11

Fire 14

Match and pop string

Fire 13

Match and pop )

Match and pop ;

Fire 18

Match and pop end

Match and pop do

Match and pop ;

Fire 17

Fire 29

Match and pop read

Match and pop (

Fire 11

Fire 14

Match and pop string

Fire 13

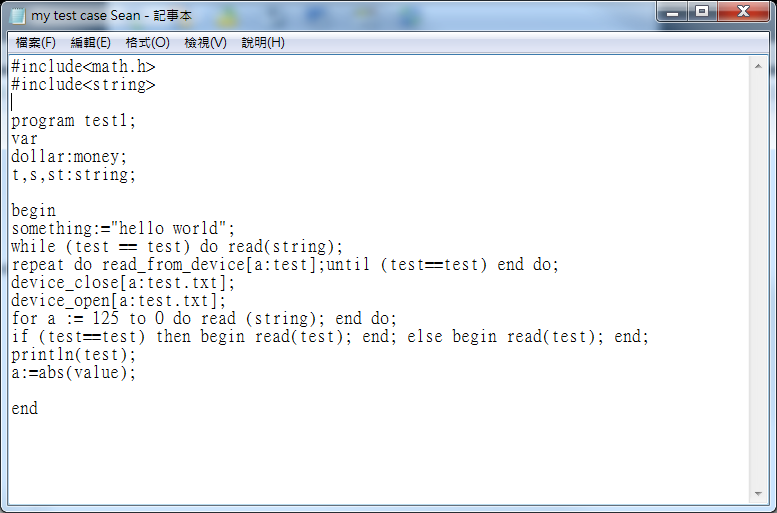
Match and pop )

Match and pop ;

Match and pop end

Accept!!

# Input of My\_test\_case:



# Result of My\_test\_case :

-----------------------Parser-----------------

Fire 1

Fire 5

Match and pop #include<math.h>

Fire 3

Fire 5

Match and pop #include<string>

Fire 4

Fire 6

Fire 7

Match and pop program

Match and pop test1

Match and pop ;

Fire 8

Match and pop var

Fire 11

Fire 14

Match and pop dollar

Fire 13

Match and pop :

Match and pop money

Match and pop ;

Fire 9

Fire 11

Fire 14

Match and pop t

Fire 12

Match and pop ,

Fire 14

Match and pop s

Fire 12

Match and pop ,

Fire 14

Match and pop st

Fire 13

Match and pop :

Match and pop string

Match and pop ;

Fire 10

Fire 15

Match and pop begin

Fire 16

Fire 19

Match and pop something

Match and pop :=

Fire 38

Match and pop "hello world"

Match and pop ;

Fire 17

Fire 23

Match and pop while

Match and pop (

Fire 30

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Fire 35

Match and pop ==

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Match and pop )

Match and pop do

Fire 16

Fire 29

Match and pop read

Match and pop (

Fire 11

Fire 14

Match and pop string

Fire 13

Match and pop )

Match and pop ;

Fire 17

Fire 22

Match and pop repeat

Match and pop do

Fire 16

Fire 26

Match and pop read\_from\_device

Match and pop [a:test]

Match and pop ;

Fire 18

Match and pop until

Match and pop (

Fire 30

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Fire 35

Match and pop ==

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Match and pop )

Match and pop end

Match and pop do

Match and pop ;

Fire 17

Fire 25

Match and pop device\_close

Match and pop [a:test.txt]

Match and pop ;

Fire 20

Match and pop device\_open

Match and pop [a:test.txt]

Match and pop ;

Fire 17

Fire 21

Match and pop for

Match and pop a

Match and pop :=

Match and pop 125

Match and pop to

Match and pop 0

Match and pop do

Fire 16

Fire 29

Match and pop read

Match and pop (

Fire 11

Fire 14

Match and pop string

Fire 13

Match and pop )

Match and pop ;

Fire 18

Match and pop end

Match and pop do

Match and pop ;

Fire 17

Fire 24

Match and pop if

Match and pop (

Fire 30

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Fire 35

Match and pop ==

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Match and pop )

Match and pop then

Match and pop begin

Fire 16

Fire 29

Match and pop read

Match and pop (

Fire 11

Fire 14

Match and pop test

Fire 13

Match and pop )

Match and pop ;

Fire 18

Match and pop end

Match and pop ;

Match and pop else

Match and pop begin

Fire 16

Fire 29

Match and pop read

Match and pop (

Fire 11

Fire 14

Match and pop test

Fire 13

Match and pop )

Match and pop ;

Fire 18

Match and pop end

Match and pop ;

Fire 17

Fire 28

Match and pop println

Match and pop (

Fire 37

Fire 42

Fire 47

Match and pop test

Fire 45

Fire 41

Match and pop )

Match and pop ;

Fire 17

Fire 19

Match and pop a

Match and pop :=

Fire 37

Fire 42

Fire 52

Match and pop abs

Match and pop (

Fire 37

Fire 42

Fire 47

Match and pop value

Fire 45

Fire 41

Match and pop )

Fire 45

Fire 41

Match and pop ;

Match and pop end

Accept!!