



40-414 Compiler Design

Semantic Analysis & Symbol Table Management

Lecture 7

Exercise

Question?

Which one of the modules detects the error in the given Pascal piece of code, and when?

```
type a = array[1..10] of integer;  
var i : integer; b : a;  
i := 11;  
b[i] = 25;
```

Lexical Analysis
in Compile time

Semantic Analysis
in Compile time

Syntax Analysis
in Compile time

Generated Code
in Runtime

Answer!

Which one of the modules detects the error in the given Pascal piece of code, and when?

```
type a = array[1..10] of integer;  
var i : integer; b : a;  
i := 11;  
b[i] = 25;
```

Lexical Analysis
in Compile time

Syntax Analysis
in Compile time

Semantic Analysis
in Compile time

Generated Code
in Runtime

Question?

What is the state of symbol table and scope stack at the time of compiling lines 7 and 13?

```
1  Program S()
2      Var a[1..5], c, real
3      Procedure R(m: integer)
4          Var b[1..5] integer
5          Procedure E()
6              Var I, c[1..3] integer
7              c(3) := a(2) + b(1)
8          End E
9      Function Q(n: integer): integer
10         Var a integer
11         Procedure P()
12             Var b real
13             b := a + c
14         End P
15     End Q
16 End R
17 End S
```

Answer!



	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10        Var a integer
11        Procedure P()
12            Var b real
13            b := a + c
14        End P
15    End Q
16    End R
17End S
  
```

Answer! (Cont.)



	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3					
4					
5					
6					
7					
8					
9					
10					
11					

```

1 Program S()
2   Var a[1..5], c, real
3   Procedure R(m: integer)
4     Var b[1..5] integer
5     Procedure E()
6       Var I, c[1..3] integer
7       c(3) := a(2) + b(1)
8     End E
9   Function Q(n: integer): integer
10     Var a integer
11     Procedure P()
12       Var b real
13       b := a + c
14     End P
15   End Q
16 End R
17 End S
  
```

Answer! (Cont.)



	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4					
5					
6					
7					
8					
9					
10					
11					

```

1 Program S()
2   Var a[1..5], c, real
3   Procedure R(m: integer)
4     Var b[1..5] integer
5     Procedure E()
6       Var I, c[1..3] integer
7       c(3) := a(2) + b(1)
8     End E
9   Function Q(n: integer): integer
10     Var a integer
11     Procedure P()
12       Var b real
13       b := a + c
14     End P
15   End Q
16 End R
17 End S
  
```

Answer! (Cont.)

Scope stack
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6					
7					
8					
9					
10					
11					

```

1 Program S()
2   Var a[1..5], c, real
3   Procedure R(m: integer)
4     Var b[1..5] integer
5     Procedure E()
6       Var I, c[1..3] integer
7       c(3) := a(2) + b(1)
8     End E
9   Function Q(n: integer): integer
10    Var a integer
11    Procedure P()
12      Var b real
13      b := a + c
14    End P
15  End Q
16 End R
17End S
  
```


Answer! (Cont.)

Scope stack
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7					
8					
9					
10					
11					

```

1 Program S()
2   Var a[1..5], c, real
3   Procedure R(m: integer)
4     Var b[1..5] integer
5     Procedure E()
6       Var I, c[1..3] integer
7       c(3) := a(2) + b(1)
8     End E
9   Function Q(n: integer): integer
10    Var a integer
11    Procedure P()
12      Var b real
13      b := a + c
14    End P
15  End Q
16 End R
17End S
  
```

Answer! (Cont.)

Scope stack
7
4
0

	Lexeme	proc/func/var	No. Arg/Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	I	var	0	int	3
8	c	array	3	int	3
9					
10					
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10        Var a integer
11        Procedure P()
12            Var b real
13            b := a + c
14        End P
15    End Q
16    End R
17End S
    
```

Answer! (Cont.)

Scope stack
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	Q	func			2
8					
9					
10					
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10         Var a integer
11         Procedure P()
12             Var b real
13             b := a + c
14         End P
15     End Q
16 End R
17 End S
    
```

Answer! (Cont.)

Scope stack
8
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	Q	func	1	int	2
8	n	param	0	int	3
9					
10					
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10         Var a integer
11         Procedure P()
12             Var b real
13             b := a + c
14         End P
15     End Q
16 End R
17 End S
  
```

Answer! (Cont.)

Scope stack
8
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	Q	func	1	int	2
8	n	param	0	int	3
9	a	var	0	int	3
10	P	proc	0	-	3
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10        Var a integer
11        Procedure P()
12            Var b real
13            b := a + c
14        End P
15    End Q
16    End R
17End S
  
```

Answer! (Cont.)

Scope stack
11
8
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	Q	func	1	int	2
8	n	param	0	int	3
9	a	var	0	int	3
10	P	proc	0	-	3
11	b	var	0	real	4

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10        Var a integer
11        Procedure P()
12            Var b real
13            b := a + c
14        End P
15    End Q
16    End R
17End S
  
```

Answer! (Cont.)

Scope stack
8
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	Q	func	1	int	2
8	n	param	0	int	3
9	a	var	0	int	3
10	P	proc	0	-	3
11					

```

1 Program S()
2   Var a[1..5], c, real
3   Procedure R(m: integer)
4     Var b[1..5] integer
5     Procedure E()
6       Var I, c[1..3] integer
7       c(3) := a(2) + b(1)
8     End E
9   Function Q(n: integer): integer
10    Var a integer
11    Procedure P()
12      Var b real
13      b := a + c
14    End P
15  End Q
16  End R
17End S
  
```

Answer! (Cont.)

Scope stack
4
0

	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4	m	param	0	int	2
5	b	array	5	int	2
6	E	proc	0	-	2
7	Q	func	1	int	2
8					
9					
10					
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10        Var a integer
11        Procedure P()
12            Var b real
13            b := a + c
14        End P
15    End Q
16    End R
17End S
  
```


Answer! (Cont.)



	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0	S	proc	0	-	1
1	a	array	5	real	1
2	c	var	0	real	1
3	R	proc	0	-	1
4					
5					
6					
7					
8					
9					
10					
11					

```

1 Program S()
2   Var a[1..5], c, real
3   Procedure R(m: integer)
4     Var b[1..5] integer
5     Procedure E()
6       Var I, c[1..3] integer
7       c(3) := a(2) + b(1)
8     End E
9   Function Q(n: integer): integer
10    Var a integer
11    Procedure P()
12      Var b real
13      b := a + c
14    End P
15  End Q
16 End R
17 End S
  
```

Answer! (Cont.)



	Lexeme	proc/ func/ var	No. Arg/ Cell	type	scope
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

```

1 Program S()
2     Var a[1..5], c, real
3     Procedure R(m: integer)
4         Var b[1..5] integer
5         Procedure E()
6             Var I, c[1..3] integer
7             c(3) := a(2) + b(1)
8         End E
9     Function Q(n: integer): integer
10        Var a integer
11        Procedure P()
12            Var b real
13            b := a + c
14        End P
15    End Q
16    End R
17End S
  
```

Note: A red arrow points from the text "End R" on line 16 to the "End S" on line 17.