

1. קיימת פרוצדורה Main בקוד והיא יחידה.

```
proc goo() {  
    proc foo() {  
        var c2 : char;  
        var a1, b : string[1000];  
        var x : int;  
    }  
}  
  
proc Main() {  
    proc foo() {  
        var c2 : char;  
        var a1, b : string[1000];  
        var x : int;  
    }  
}
```

```
1 Error!, must be at least 1 procedure of 'Main' in the code!  
2  
  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6 BODY  
7 PROC  
8 foo  
9 ARGS NONE  
10  
11  
12  
13  
14 BODY DEC  
15 c2 : char  
16 a1 NEXT b : string [ ATYPE 1000 ] DEC  
17 x : int
```

2. Main לא מקבל ארגומנטים

```
1 proc Main(x, y : int) {  
2  
3     proc foo() {  
4         var c2 : char;  
5         var a1, b : string[1000];  
6         var x : int;  
7     }  
8  
9 }
```

```
1 Error!, No arguments to 'Main' is allowed!  
2
```

3. לא קיימות שתי פונקציות או פרוצדורות עם אותו שם באותו scope

```
proc Main() {  
    proc goo() {  
        var a : int;  
    }  
    proc foo() {  
        var c2 : char;  
        var a1, b : string[1000];  
        var x : int;  
    }  
}
```

```
1 Error!, illegal Procedure! foo is already used, there are no two functions or procedures or variables with the same name in the same scope  
2  
  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6 BODY  
7 PROC  
8 goo  
9 ARGS NONE  
10  
11  
12 BODY DEC  
13 a : int  
14  
15 PROC  
16 foo  
17 ARGS NONE  
18  
19  
20  
21 BODY DEC  
22 c2 : char  
23 a1 NEXT b : string [ ATYPE 1000 ] DEC  
24 x : int
```

4. לא קיימים שני משתנים עם אותו שם באותו scope

```
proc Main() {  
    proc goo() {  
        var a : int;  
    }  
    proc foo() {  
        var x : int;  
        var a1, b : string[1000];  
        var x1 : int;  
    }  
}
```

```
1 Error!, illegal Variable! x is already used, there are no two functions or procedures or variables with the same name in the same scope  
2  
  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6 BODY  
7 PROC  
8 goo  
9 ARGS NONE  
10  
11  
12 BODY DEC  
13 a : int  
14  
15 PROC  
16 foo  
17 ARGS NONE  
18  
19  
20  
21 BODY DEC  
22 x : int  
23 a1 NEXT b : string [ ATYPE 1000 ] DEC  
24 x1 : int
```

5. פונקציות ופרוצדורות הוגדרו לפני שמפעילים אותן.

```
proc Main() {  
  proc goo() {  
    var a : int;  
  }  
  foo();  
}  
  
proc Main() {  
  proc goo() {  
    var a : int;  
  }  
  proc foo() {  
    var x : int;  
  }  
  foo();  
}
```

```
1 Error! foo is not define before use!  
2  
3  
  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6  
7 BODY  
8 PROC  
9 goo  
10 ARGS NONE  
11  
12 BODY  
13 DEC  
14 | a : int  
15  
16 PROC  
17 foo  
18 ARGS NONE  
19  
20 BODY  
21 DEC  
22 | x : int  
23 | EXPR CALL foo  
24
```

6. משתנים הוגדרו לפני שמשתמשים בהם.

```
proc Main() {  
  proc goo() {  
    a=5;  
  }  
  proc foo() {  
    var x : int;  
  }  
}  
  
proc Main() {  
  proc goo() {  
    var a : int;  
    a=5;  
  }  
  proc foo() {  
    var x : int;  
  }  
}
```

```
1 Error! a is not define before use!  
2  
3  
  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6 BODY  
7 PROC  
8 goo  
9 ARGS NONE  
10  
11 BODY  
12 DEC  
13 | a : int EXPR = ATYPE USEID a ATYPE 5  
14  
15 PROC  
16 foo  
17 ARGS NONE  
18  
19 BODY  
20 DEC  
21 | x : int  
22
```

7. כמות הארגומנטים בקריאה לפונקציה/פרוצדורה צריכה להיות שווה לכמות הארגומנטים בהגדרת הפונקציה/פרוצדורה.

```
proc Main() {  
  proc goo() {  
    var a : int;  
    a=5;  
  }  
  proc foo(a,b,c : int) {  
    var x : int;  
  }  
  foo[3];  
}  
  
proc Main() {  
  proc goo() {  
    var a : int;  
    a=5;  
  }  
  proc foo(a,b,c : int) {  
    var x : int;  
  }  
  foo[3,4,5];  
}
```

```
1 Error! The number of arguments passed to 'foo' is not as define!  
2  
3  
  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6 BODY  
7 PROC  
8 goo  
9 ARGS NONE  
10  
11 BODY  
12 DEC  
13 | a : int EXPR = ATYPE USEID a ATYPE 5  
14  
15 PROC  
16 foo  
17 STARGS  
18 ARGS  
19 | a NEXTP b NEXTP c : int
```

8. טיפוסים של הארגומנטים בקריאה לפונקציה/פרוצדורה תואמים לטיפוסים בהגדרת הפונקציה/פרוצדורה.

```
proc Main() {
  proc foo(a,b,c : int; x,y : real; f : bool) {
    var x : int;
    foo(3,4,5,5,5,5);
  }
}

proc Main() {
  proc foo(a,b,c : int; x,y : real; f : bool) {
    var x : int;
    foo(3,4,5,5,5,5,5);
  }
}

proc Main() {
  proc foo(a,b,c : int; x,y : real; f : bool) {
    var x : int;
    foo(3,4,5,5,5,5,true);
  }
}
```

```
1 Error! foo expect argument of type real but given argument has type of int
2
3
4

1 Error! foo expect argument of type bool but given argument has type of int
2
3
4

1 CODE
2
3 PROC
4 Main
5 ARGV NONE
6
7 BODY
8 PROC
9 foo STARGS
10 ARGV
11 a NEXT b NEXT c : int ; STARGS
12 ARGV
13 x NEXT y : real ; STARGS
14 ARGV
15 f : bool
16 BODY
17 x : int
18 DEC
```

9. טיפוס הערך המוחזר מהפונקציה תואם לטיפוס ההחזרה המוכרז בכותרת של הפונקציה. וטיפוס ההחזרה של הפונקציה לא יכול להיות מחרוזת.

```
proc Main() {
  func foo(a : int) return int {
    return true;
  }
}

proc Main() {
  func foo(a : int) return int {
    return 0;
  }
}
```

```
1 Error! The Function foo return type is int but the return value has type of bool
2
3
4

1 CODE
2
3 PROC
4 Main
5 ARGV NONE
6
7 BODY
8 FUNC
9 foo STARGS
10 ARGV
11 a : int
12
13 RET
14 int
15
16 RET
17 0 int
18
```

10. טיפוס הערך המוחזר מהפונקציה תואם לטיפוס המשתנה שלתוכו נכנס הערך שמוחזר מהפונקציה.

```
proc Main() {
  func foo(a : int) return int {
    return 0;
  }

  var a : char;
  a = foo(3);
}

proc Main() {
  func foo(a : int) return int {
    return 0;
  }

  var a : int;
  a = foo(3);
}
```

```
1 Error! Function foo type is 'int' but 'a' has type of 'char'
2
3
4

1 CODE
2
3 PROC
4 Main
5 ARGV NONE
6
7 BODY
8 FUNC
9 foo STARGS
10 ARGV
11 a : int
12
13 RET
14 int
15
16 RET
17 0 int
18 DEC
19 a : int    EXPR = ATYPE USEID a  CALL foo ATYPE ATYPE
20 int
21 int
```

11. טיפוס התנאי ב if -הוא מטיפוס bool
12. טיפוס התנאי ב while -וב for -הוא מטיפוס bool

```
proc Main() {
  var x : int;
  if (x+5) {
    var y: int;
    y=3;
  }
}

proc Main() {
  var x : int;
  if (x>5) {
    var y: int;
    y=3;
  }
}

proc Main() {
  var x : int;
  while (x+8) {
    var z: int;
    z=3;
  }
}

proc Main() {
  var x : int;
  while (true) {
    var z: int;
    z=3;
  }
}

proc Main() {
  var x : int;
  for (int i=0; i+3 ;i+1) {
    var w: int;
    w=3;
  }
}

proc Main() {
  var x : int;
  for (int i=0; i<3 ;i+1) {
    var w: int;
    w=3;
  }
}
```

```
1 Error! The Condition in 'IF' statment must be of boolean type
2
3
4

1 CODE
2 PROC
3   Main
4   ARGS NONE
5
6   BODY      DEC
7     x : int  IF
8     COND > ATYPE USEID x  ATYPE 5
9
10    BLOCK
11      DEC
12      y : int  EXPR = ATYPE USEID y  ATYPE 3
13
14
15

1 Error! The Condition in 'WHILE' statment must be of boolean type
2
3
4

1 CODE
2 PROC
3   Main
4   ARGS NONE
5
6   BODY      DEC
7     x : int  WHILE COND ATYPE true bool
8
9
10    BLOCK
11      DEC
12      z : int  EXPR = ATYPE USEID z  ATYPE 3
13
14
15

1 Error! The Condition in 'FOR' statment must be of boolean type
2
3
4

1 CODE
2 PROC
3   Main
4   ARGS NONE
5
6   BODY      DEC
7     x : int
8     FOR
9     int = i COND < ATYPE USEID i  ATYPE 3 + ATYPE USEID i  ATYP
10    BLOCK
11      DEC
12      w : int  EXPR = ATYPE USEID w  ATYPE 3
13
14
15
```

13. טיפוס הביטוי המופיע כאינדקס ב-[] של מחרוזת הוא מטיפוס int.
14. לא משתמשים באופרטור [] בשום טיפוס חוץ ממחרוזת.

```
proc Main() {
  var x : char*;
  var y['d']: string;
}

proc Main() {
  var x[0] : char*;
  var y[5]: string;
}

proc Main() {
  var x : char*;
  var y[5]: string;
}
```

```
1 Error! The index parameter inside '[' place must be of type 'int'
2
3
4

1 Error! The '[' operator is available only for 'string' type
2
3
4

1 CODE
2 PROC
3   Main
4   ARGS NONE
5
6   BODY      DEC
7     x : char*  DEC
8     y : string [ ] [ ATYPE 5 int ]
9
10
```

15. טיפוס של המשתנה מצד שמאל של האופרטור השמה (=) תואם לטיפוס הביטוי מצד ימין.

<pre>proc Main() { var a : int; a = 5.5; }</pre>	<pre>1 Error! Variable 'a' define type is 'int' but assign value has type of 'real' 2 3 4</pre>
<pre>proc Main() { var b : char; b = 5; }</pre>	<pre>1 Error! Variable 'b' define type is 'char' but assign value has type of 'int' 2 3 4</pre>
<pre>proc Main() { var c : real; c = 'd'; }</pre>	<pre>1 Error! Variable 'c' define type is 'real' but assign value has type of 'char' 2 3 4</pre>
<pre>proc Main() { var d : string; d = true; }</pre>	<pre>1 Error! Variable 'd' define type is 'string' but assign value has type of 'bool' 2 3 4</pre>
<pre>proc Main() { var e : bool; e = "hhfgh"; }</pre>	<pre>1 Error! Variable 'e' define type is 'bool' but assign value has type of 'string' 2 3 4</pre>
<pre>proc Main() { var a : int; var b : char; var c : real; var d : string; var e : bool; a = 5; b = 'g'; c = 5.5; d = "gffgf"; e = true; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 a : int DEC 8 b : char DEC 9 c : real DEC 10 d : string DEC 11 e : bool EXPR = ATYPE USEID a ATYPE 5 int EXPR = ATYPE USEID 12 13</pre>

16. עבור אופרטורים (+, -, *, /) שני האופרנדים חייבים להיות int או real. והתוצאה היא int, אם שני האופרנדים int, אחרת הטיפוס של הביטוי הוא real.

<pre>proc Main() { var a : int; var b : real; b = 5+5; a = 5.5+5; }</pre>	<pre>1 Error! Variable 'b' type is 'real' and the result of '+' operator is of type 'int' 2 3 4</pre>
<pre>proc Main() { var a : int; var b : real; b = 5+5.3; a = 5+5; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 Error! Variable 'b' type is 'real' and the result of '+' operator is of type 'int' 8 9</pre>
<pre>proc Main() { var a : int; var b : real; b = 5 - 5; a = 5 - 5.5; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 Error! Variable 'b' type is 'real' and the result of '-' operator is of type 'int' 8 9</pre>
<pre>proc Main() { var a : int; var b : real; b = 5 - 3.5; a = 5 - 5; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 Error! Variable a type is 'int' but assign value has type of 'real' 8 9</pre>
<pre>proc Main() { var a : int; var b : real; a = 5.2 * 5; b = 5 * 35; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 Error! Variable a type is 'int' but assign value has type of 'real' 8 9</pre>
<pre>proc Main() { var a : int; var b : real; a = 5 * 5; b = 5 * 3.5; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 Error! Variable a type is 'int' but assign value has type of 'real' 8 9</pre>
<pre>proc Main() { var a : int; var b : real; a = 5.5 / 5; b = 5 / 35; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 Error! Variable a type is 'int' but assign value has type of 'real' 8 9</pre>
<pre>proc Main() { var a : int; var b : real; a = 55 / 5; b = 5 / 3.5; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 BODY DEC 7 a : int DEC 8 b : real EXPR = ATYPE USEID a / ATYPE 55 int ATYPE 5 int EXPR = ATYPE 9 10</pre>

17. עבור אופרטורים (&&, ||) שני האופרנדים חייבים להיות bool והתוצאה היא bool.

<pre>proc Main() { var a : bool; var b : bool; a = 55 && 5; }</pre>	<pre>1 Error! '&&' operator is available for type 'bool' only! not 'int' 2 3 4</pre>
<pre>proc Main() { var a : bool; var b : bool; a = true && b; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6</pre>
<pre>proc Main() { var a : bool; var b : bool; a = true 'h'; }</pre>	<pre>1 Error! ' ' operator is available for type 'bool' only! not 'char' 2 3 4</pre>
<pre>proc Main() { var a : bool; var b : bool; a = true false; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : bool DEC 9 b : bool EXPR = ATYPE USEID a ATYPE true bool ATYPE 10</pre>

18. עבור האופרטורים (>, <=, <, >=) שני האופרנדים חייבים להיות int או real והתוצאה היא bool

<pre>proc Main() { var a : int; var b : bool; var c : real; b = c > 5; b = 5 > 35; }</pre>	<pre>1 Error! '>' operator is available only for same type 'int' or 'real' 2 3 4</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = 2 > 5; b = 5 > 35; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : int DEC 9 b : bool DEC 10 c : real EXPR = ATYPE USEID b > ATYPE 2 int ATYPE 5 int EXPR 11</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = 'u' >= 'h'; b = 5 >= 35; }</pre>	<pre>1 Error! '>=' operator is available for type 'int' or 'real' only! not 'char' 2 3 4</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = 5.5 >= 2.3; b = 5 >= 35; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : int DEC 9 b : bool DEC 10 c : real EXPR = ATYPE USEID b >= ATYPE 5.5 real ATYPE 2.3 real 11</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = true < false; b = 5 < 35; }</pre>	<pre>1 Error! '<' operator is available for type 'int' or 'real' only! not 'bool' 2 3 4</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = 4.2 < 4.3; b = 5 < 35; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : int DEC 9 b : bool DEC 10 c : real EXPR = ATYPE USEID b < ATYPE 4.2 real ATYPE 4.3 real 11</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = b <= b; b = 5 <= 35; }</pre>	<pre>1 Error! '<=' operator is available for type 'int' or 'real' only! not 'bool' 2 3 4</pre>
<pre>proc Main() { var a : int; var b : bool; var c : real; b = a <= a; b = 5 <= 3; }</pre>	<pre>1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : int DEC 9 b : bool DEC 10 c : real EXPR = ATYPE USEID b <= ATYPE USEID a ATYPE USEID a 11</pre>

19. עבור האופרטורים (\neq , $=$) שני האופרנדים יכולים להיות שני `int/real/bool/char/pointers` והתוצאה היא `bool`.

<pre> proc Main() { var a : int; var b : bool; var c : real; b = a != a; b = 5 != c; } proc Main() { var a : int; var b : bool; var c : real; b = 's' != 's'; b = 5.20 != 1.2; } proc Main() { var a : int; var b : bool; var c : real; b = "sdds" == 3; b = 5.20 == 1.2; } proc Main() { var a : int*; var b : bool; var c : int*; b = 50 == 30; b = 5.20 == 1.2; } </pre>	<pre> 1 Error! '!=' operator is available for same types only! 2 3 4 1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : int DEC 9 b : bool DEC 10 c : real EXPR = ATYPE USEID b != ATYPE 's' char 11 1 Error! '==' operator is available only for same types 2 3 4 1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : int* DEC 9 b : bool DEC 10 c : int* EXPR = ATYPE USEID b == ATYPE 50 int AT 11 </pre>
--	---

20. אופרטור ערך מוחלט (`||`) יכול להיות מופעל על מחרוזות והתוצאה היא `int`.

<pre> proc Main() { var a : string; var b : int; b = 'd' ; } proc Main() { var a : string; var b : int; b = a ; } </pre>	<pre> 1 Error! 'ABS' operator is available for string type only! not 'char' 2 3 4 1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : string DEC 9 b : int EXPR = ATYPE USEID b ATYPE ABS ATYPE USEID a 10 </pre>
---	--

21. אופרטור (!) יכול להיות מופעל על `bool` והתוצאה היא `bool`.

<pre> proc Main() { var a : bool; var b : int; a = !b; } proc Main() { var a : bool; var b : int; a = !true; } </pre>	<pre> 1 Error! '!' operator is available for type 'bool' only! not 'int' 2 3 4 1 CODE 2 PROC 3 Main 4 ARGS NONE 5 6 7 BODY DEC 8 a : bool DEC 9 b : int EXPR = ATYPE USEID a ! true bool 10 </pre>
--	---

22. אופרטור & מופעל רק על משתנים מטיפוס int/real/char/string תוצאה למצביע

```
proc Main() {  
  var a : int;  
  var b: real;  
  a = &b;  
}  
proc Main() {  
  var a : int*;  
  var b: bool;  
  a = &b;  
}  
proc Main() {  
  var a : int*;  
  var b: real;  
  a = &b;  
}
```

```
1 Error! Variable 'a' define type is 'int' but assign value has address for pointers only!  
2  
3  
4  
1 b  
2 Error! '&' operator is available for types only!  
3  
4  
5  
1 b  
2 CODE  
3 PROC  
4 Main  
5 ARGS NONE  
6  
7  
8 BODY DEC  
9 a : int* DEC  
10 b : real EXPR = ATYPE USEID a ATYPE & b  
11
```

23. אופרטור ^ מופעל רק על מצביעים

```
proc Main() {  
  var a : int*;  
  var b: bool;  
  a = ^b;  
}  
proc Main() {  
  var a : int;  
  var b: bool;  
  a = ^b;  
}  
proc Main() {  
  var a : int;  
  var b: real*;  
  a = ^b;  
}
```

```
1 Error! Variable 'a' define type is 'int*' but assign value of pointers to variable only!  
2  
3  
4  
1 Error! '^' operator is available for pointers only!  
2  
3  
4  
1 CODE  
2 PROC  
3 Main  
4 ARGS NONE  
5  
6  
7 BODY DEC  
8 a : int DEC  
9 b : real* EXPR = ATYPE USEID a ATYPE ^ b  
10
```