פרוייקט בקומפילציה | חלק 2

אסף ארדיטי - 311533640 | תאיר חדד 204651897| יונתן מכלוף 308038256

דוגמאות הרצה:

1. קיימת פונקציה MAIN בקוד והיא יחידה:

```
function int foo(int i)
{
    return 1;
}
```

```
Where is your main?!?!?!
```

```
function int foo(int i)
{
    return 1;
}

function void main()
{
    var int x;
    return;
}
function void main()
{
    var int x;
    return;
}
```

```
Function main already declared
```

2. Main לא מקבל ארגומנטים והטיפוס שלה void:

```
function int main()
{
    var int x;
    return 1;
}
```

```
main must be void type
```

```
function void main(int x)
{
    var int x;
    return;
}
```

```
main can not be with parameters
```

3. לא קיימות שתי פונקציות עם אותו שם באותו scope:

```
function int foo(int i)
{
    function int foo(int i)
    {
        return 1;
    }
    return 1;
}

function void main()
{
    var int x;
    return;
}
```

```
OK
```

```
function int foo(int i)
{
    return 1;
}
function int foo(int i)
{
    return 1;
}
function void main(int x)
{
    var int x;
    return;
}
```

```
Function foo already declared
```

4. **לא קיימים שני משתנים עם אותו שם באותו scope:**

```
function int foo(int i)
{
    var int x=5;
    var char x='c';
    return 1;
}
```

```
Variable 'x' already declared
```

5. **פונקציות הוגדרו לפני שמפעילים אותן:**

```
function void main()
{
    foo(1);
    return;
}
function int foo(int i)
{
    var int x=5;
    return 1;
}
```

```
Function 'foo' not declared
```

```
function int foo(int i)
{
    var int x=5;
    return 1;
}
function void main()
{
    foo(1);
    return;
}
```

```
OK
```

6. **משתנים הוגדרו לפני שמפעילים אותם:**

```
function int foo(int i)
{
    x=5;
    var int x;
    return 1;
}
```

```
Error: syntax error at line 4
Parser does not expect 'var'
```

7. **מספר הארגומנטים האקטואליים שווה למספר הארגומנטיים הפומרלים של הפונקציה – כמות הארגומנטים בקריאה לפונקציה צריכה להיות שווה לכמות הארגומנטים בהגדרת הפונקציה:**

```
function int foo(int i,x,z)
{
    return 1;
}
function void main()
{
    foo(1,2,1,3);
    return;
}
```

```
wrong number or types of argumentsn
in function call (in the scope of 'main')
```

```
function int foo(int i,x,z,y)
{
    return 1;
}
function void main()
{
    foo(1,2,1,3);
    return;
}
```

```
OK
```

8. **טיפוסים של הארגומנטים בקריאה לפונקציה תואמים לטיפוסים בהגדרת הפונקציה:**

```
function int foo(int i; real z)
{
    return 1;
}
function void main()
{
    foo(1,'x');
    return;
}
```

```
wrong number or types of arguments
in function call (in the scope of 'main')
```

```
function int foo(int i,x; real z)
{
    return 1;
}
function void main()
{
    foo(1,2,1.2);
    return;
}
```

```
OK
```

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

```
function int foo(int i)
{
    return 'x';
}
function void main()
{
    foo(1);
    return;
}
```

```
Return 'char' instead of 'int', check your'foo' .
```

```
function string foo(int i)
{
    return "fsdf";
}
function void main()
{
    foo(1);
    return;
}
```

```
Error: syntax error at line 1
Parser does not expect 'string'
```

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

```
function int foo(int i)
{
    return i;
}
function void main()
{
    var int x;
    x= foo(1);
    return;
}
```

```
OK
```

```
function char foo(int i)
{
    return 'c';
}
function void main()
{
    var int x;
    x= foo(1);
    return;
}
```

```
Operator '=' can only be used with identical types
```

11. **טיפוס התנאי ב-IF הוא מטיפוס BOOL:**

```
function void main()
{
    var int x=2;

    if(x>2){
        x=4;
    }
    return;
}
```

```
OK
```

```
function void main()
{
    if('x'){
        var int x;
    }
    return;
}
```

```
Condition must be boolean
```

12. **טיפוס התנאי ב-while, בdo-while ובfor הוא מטיפוס bool:**

```
function void main()
{
    var int x=2;
    var int i;

    while(x>3){
        do{
            for(i=1;i==3;i++){
                x=3;
            }
        }while(true);
    }
    return;
}
```

OK

```
function void main()
{
    var int x=2;
    var int i;

    while('x'){
        do{
            for(i=1;i==3;i++){
                x=3;
            }
        }while(true);
    }
    return;
}
```

Condition must be boolean

```
function void main()
{
    var int x=2;
    var int i;
    for(i=1;3.2;i++){
        x=3;
    }

    return;
}
```

Condition must be boolean

```
function void main()
{
    var int x=2;
    var int i;
    do{
        for(i=1;i==3;i++){
            x=3;
        }
    }while("xc");

    return;
}
```

Condition must be boolean

13. **טיפוס הביטוי המופיע כאינדקס ב-[] של מחרוזת הוא מטיפוס int.**

```
function void main()
{
    string x[3];
    return;
}
```

OK

```
function void main()
{
    string x['x'];
}
```

char Size of string must be int

14. **לא משתמשים באופרטור [] בשום טיפוס חוץ ממחרוזת.**

```
function void main()
{
    var int x;
    x[3]=5;
    return;
}
```

Index operator must be only used with string variables

15. **טיפוס של המשתנה מצד שמאל של האופרטור השמה (=) תואם לטיפוס הביטוי מצד ימין. במחרוזת מותר רק תווים, NULL יכול להיות רק מטיפוס מצביע.**

```
function void main()
{
    var char x;
    x=3;
    return;
}
```

Operator '=' can only be used with identical types

```
function void main()
{
    string x[5];
    x = "fd";
    return;
}
```

OK

```
function void main()
{
    var int x;
    x = null;
    return;
}
```

NULL can be assigned only
to int* or char* or real* variabels

```
function void main()
{
    var int* x;
    x = null;
    return;
}
```

OK

16. **טיפוסים בביטויים (expressions) הם תואמים:**

- **+ - \ *:**

```
function void main()
{
    var char x;
    x='x'/'c';
    return;
}
```

The operator '/' only compatible with int or real

```
function void main()
{
    var char x;
    x='x'+'c';
    return;
}
```

The operator '+' only compatible with int or real

- **|| ,&&:**

```
function void main()
{
    var bool x;
    x=true&&false||true;
    return;
}
```

OK

```
function void main()
{
    if(3>2&&5==4||2<=3){
        var int x;
    }
    return;
}
```

OK

- <u>:<,=<,>,=></u>

```
function void main()
{
    if(4.1<3.2 && 3>2){
        var int x;
    }
    return;
}
```

OK

```
function void main()
{
    if('x'<'s'){
        var int x;
    }
    return;
}
```

The operator '<' only compatible with int

- <u>:=!,==</u>

```
function void main()
{
    var bool x;
    x= 3!=5;
    return;
}
```

OK

```
function void main()
{
    var bool x;
    x= 3=='x';
    return;
}
```

The operator '==' only compatible with int,int*,char,char*,real,real* or boolean

- <u>:||</u>

```
function void main()
{
    string x[5];
    var int y;
    y= |x|;
    return;
}
```

OK

```
function void main()
{
    string x[5];
    var int y;
    var char z;
    z='x';
    y= |z|;
    return;
}
```

ABS only compatible with int or string

- <u>:!</u>

```
function void main()
{
    var bool y;
    if(!y){
        var int x;
    }
    return;
}
```

OK

```
function void main()
{
    var char y;
    if(!y){
        var int x;
    }
    return;
}
```

operator '!' is only compatible with boolean

17. **אפשר רק להוסיף או להפחית int ממצביע:**

```
function void main()
{
    var int* y;
    y= y - 5;
    return;
}
```

OK

18. **אופרטור & מופעל רק על משתנים מטיפוס int, char,real או string[i]:**

```
function void main()
{
    var int x;
    var int* y;
    y= &x;
    return;
}
```

OK

```
function void main()
{
    var bool x;
    var int* y;
    y= &x;
    return;
}
```

```
operator '&' is only compatible with
int,string[i],real or char
```

19. **אופרטור אונרי מופעל רק על מצביעים.**

```
function void main()
{
    var int* x;

    x = (*x)+5;
    return;
}
```

OK

```
function int foo(int x,y; real z)
{
    string a[5];
    a="asaf";
    a[2]='b';
    while(x>3){
        do{
            if(y>15){
                for(y=0;y<14;y++){
                    z= z + 3.2;
                }
            }
        }while(x>13);
    }
    return 5;
}
function void main()
{
    foo(2,4,3.2);
    return ;
}
```

OK