Project 4

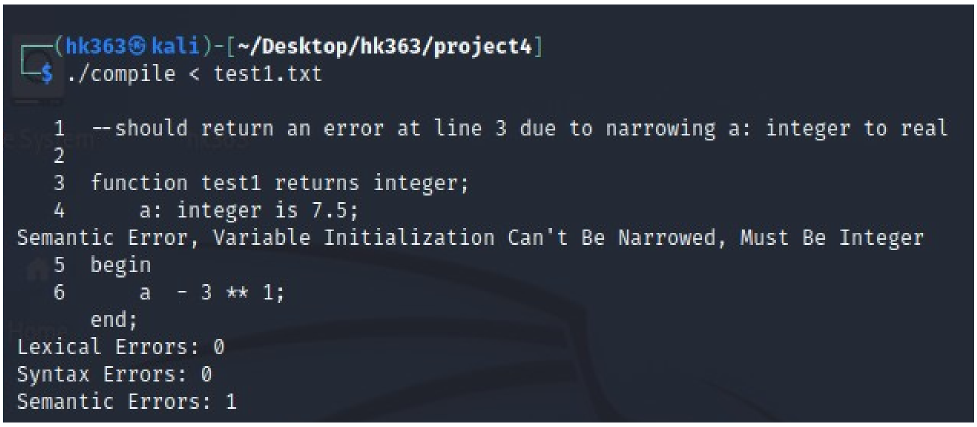
Approach to the Project:

My approach to the project was, firstly to think of what extra functions need to be created in the types.cc file in order to be able to process and do the semantic checks that need to be implemented. Then, I added the actions (and mid-rule, wherever needed) in the parser.y utilizing these functions I created in types.cc and made sure I passed the correct values to the arguments.

Then, I created the tests based on what semantic rules need to be checked.

Running the Tests:

Test 1:

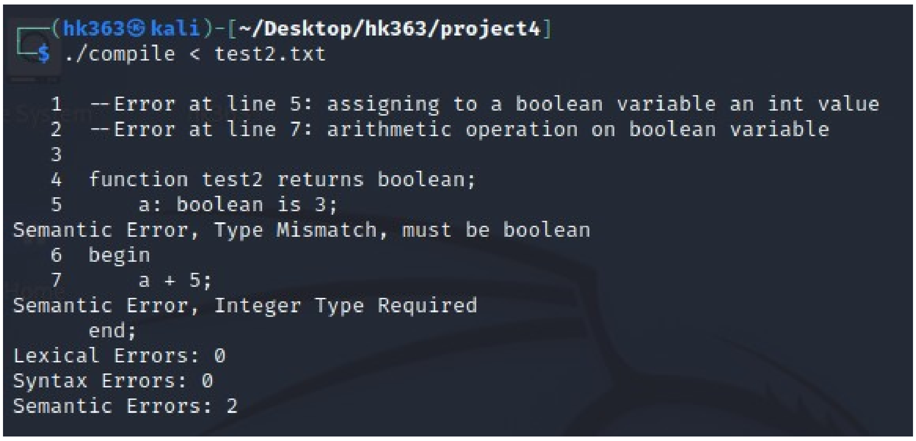


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test1.txt

   1  --should return an error at line 3 due to narrowing a: integer to real
   2
   3  function test1 returns integer;
   4      a: integer is 7.5;
Semantic Error, Variable Initialization Can't Be Narrowed, Must Be Integer
   5  begin
   6      a  - 3 ** 1;
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 1
```

The Semantic error caught here is correct, because we initially declared **a: integer** and then we assign to it a real value. This is a case of type narrowing since we move from a less precise to a more precise type, and we implemented this to be a semantic error.

Test 2:



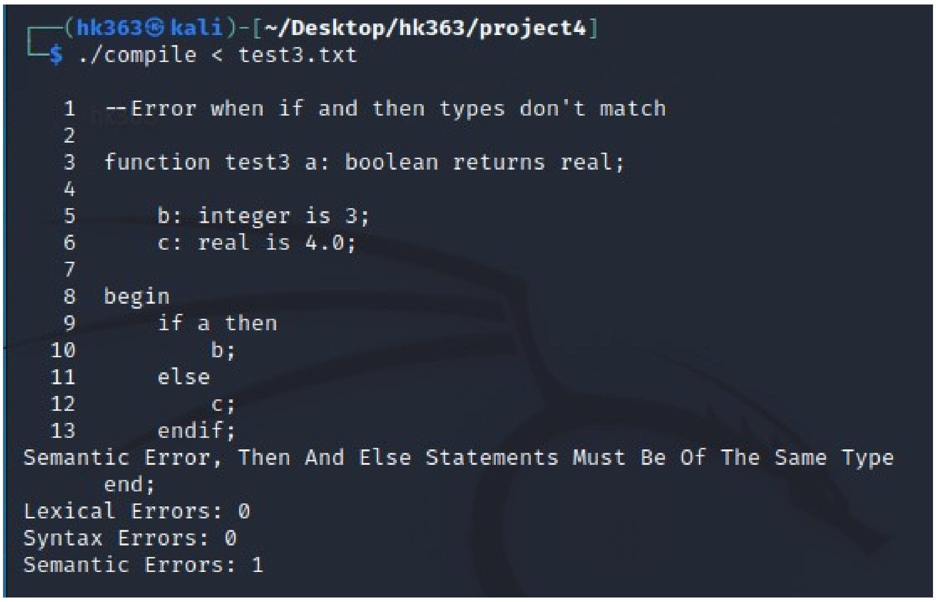
```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test2.txt

   1  --Error at line 5: assigning to a boolean variable an int value
   2  --Error at line 7: arithmetic operation on boolean variable
   3
   4  function test2 returns boolean;
   5      a: boolean is 3;
Semantic Error, Type Mismatch, must be boolean
   6  begin
   7      a + 5;
Semantic Error, Integer Type Required
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 2
```

The Semantic errors here are correctly found.

1. Line 5: We initialize a boolean variable with an integer
2. Line 7: We try to do arithmetic operation between a boolean and an int

Test 3:

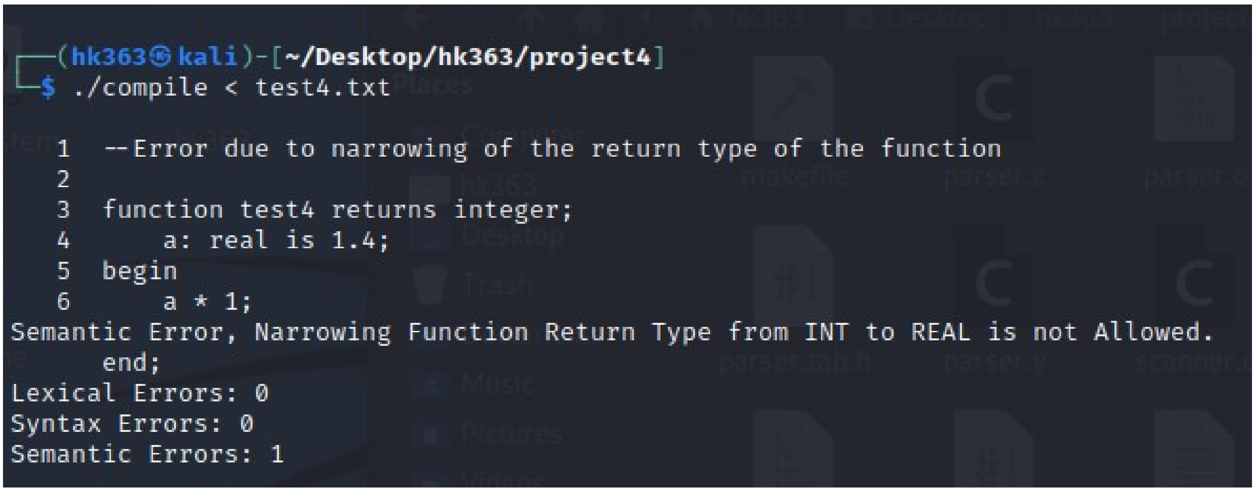


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test3.txt

   1  --Error when if and then types don't match
   2
   3  function test3 a: boolean returns real;
   4
   5      b: integer is 3;
   6      c: real is 4.0;
   7
   8  begin
   9      if a then
  10          b;
  11      else
  12          c;
  13      endif;
Semantic Error, Then And Else Statements Must Be Of The Same Type
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 1
```

The Semantic error is correctly found because the types of if and then do not match. At if branch we return an int, and at else branch we return a real. There is no function return type error,r because even if we take the if branch which returns an int, since we have declared the return type of real, there is no “type narrowing“ happening.

Test 4:

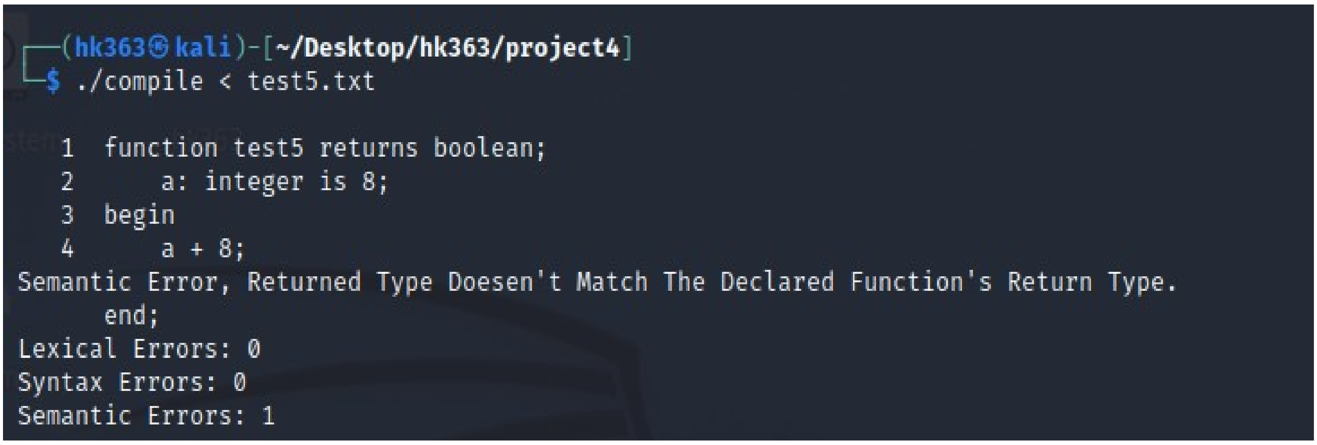


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test4.txt

   1  --Error due to narrowing of the return type of the function
   2
   3  function test4 returns integer;
   4      a: real is 1.4;
   5  begin
   6      a * 1;
Semantic Error, Narrowing Function Return Type from INT to REAL is not Allowed.
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 1
```

The Semantic error is correctly found because the function is declared to return **integer** type, but the type returned is **real**, so this is type narrowing and is not allowed.

Test 5:

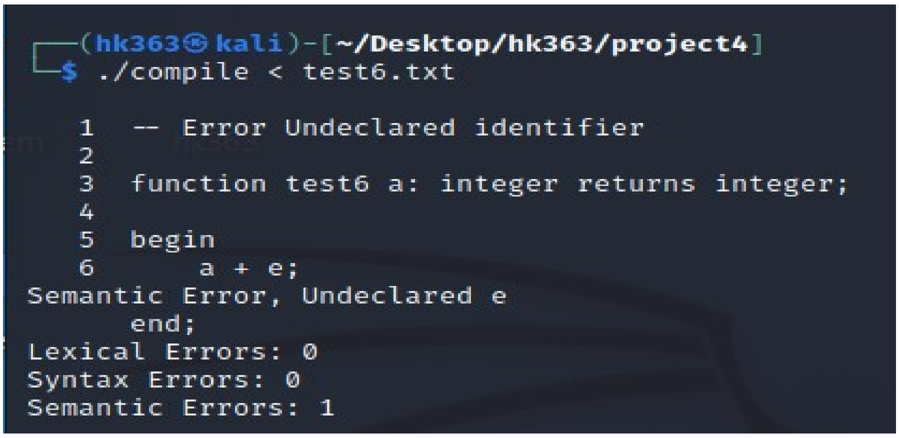


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test5.txt

   1  function test5 returns boolean;
   2      a: integer is 8;
   3  begin
   4      a + 8;
Semantic Error, Returned Type Doesen't Match The Declared Function's Return Type.
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 1
```

The Semantic error is correctly found because the function is declared to return **boolean** type, but the type returned is **int**.

Test 6:

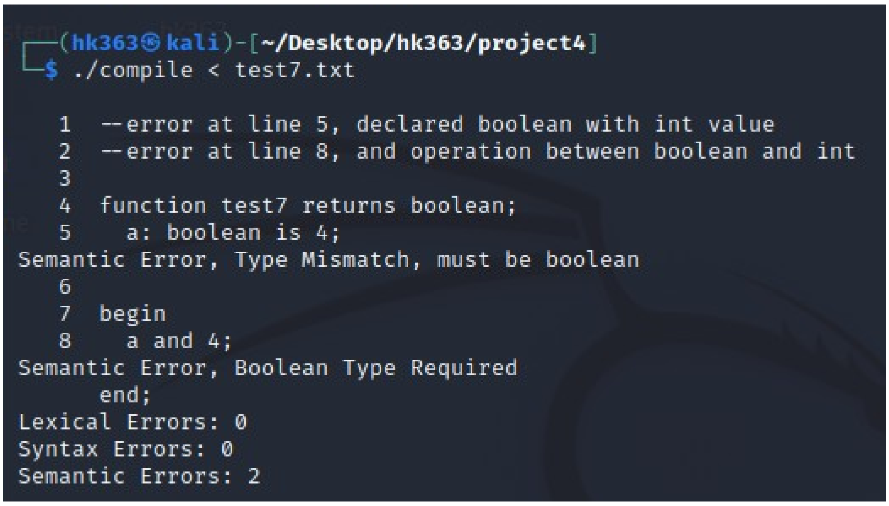


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test6.txt

   1  -- Error Undeclared identifier
   2
   3  function test6 a: integer returns integer;
   4
   5  begin
   6      a + e;
Semantic Error, Undeclared e
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 1
```

The Semantic error is correctly found because the identifier e is not declared.

Test 7:



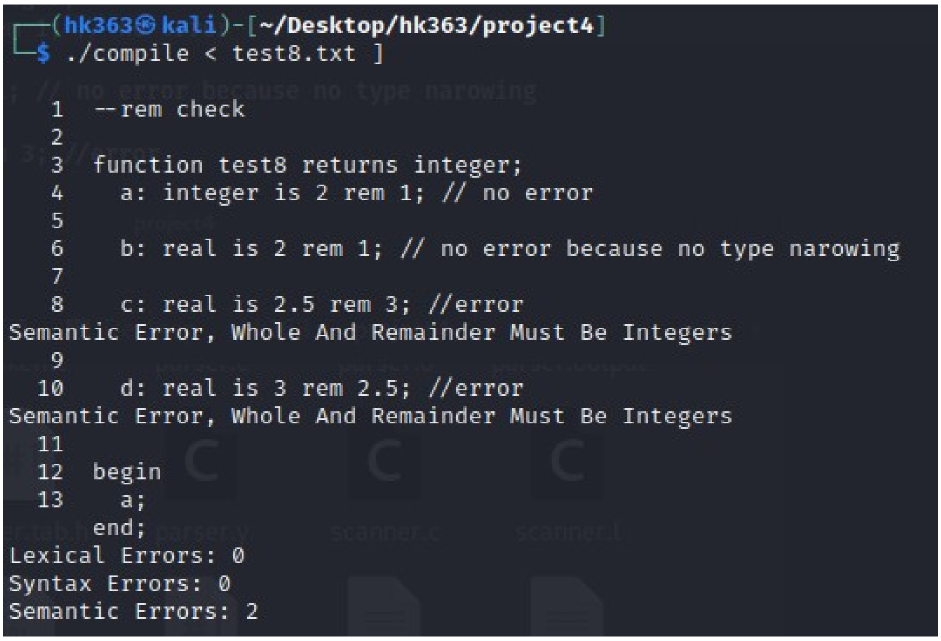
```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test7.txt

   1  --error at line 5, declared boolean with int value
   2  --error at line 8, and operation between boolean and int
   3
   4  function test7 returns boolean;
   5    a: boolean is 4;
Semantic Error, Type Mismatch, must be boolean
   6
   7  begin
   8    a and 4;
Semantic Error, Boolean Type Required
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 2
```

The Semantic errors are correctly found:

1. Line 5: We declare variable **a: boolean** and then we assign an int value to it.
2. Line 8: We do an ‘and’ operation with a boolean and an int.

Test 8:

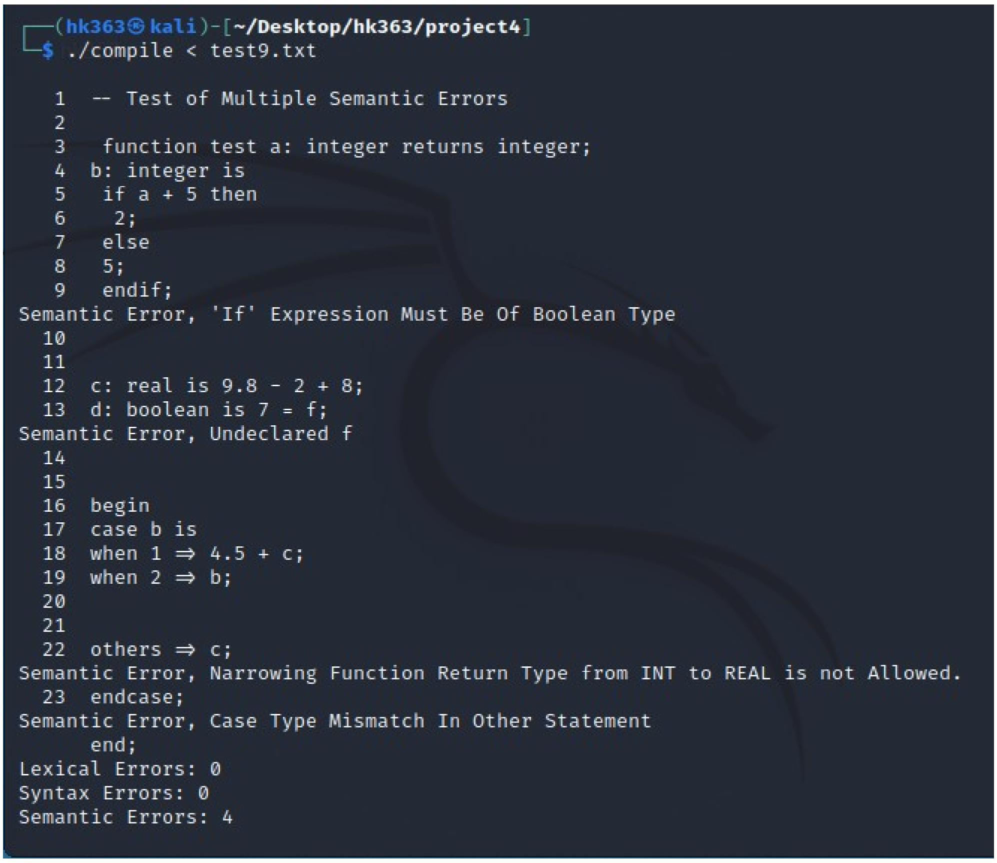


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test8.txt ]

   1  --rem check
   2
   3  function test8 returns integer;
   4    a: integer is 2 rem 1; // no error
   5
   6    b: real is 2 rem 1; // no error because no type narowing
   7
   8    c: real is 2.5 rem 3; //error
Semantic Error, Whole And Remainder Must Be Integers
   9
  10    d: real is 3 rem 2.5; //error
Semantic Error, Whole And Remainder Must Be Integers
  11
  12  begin
  13    a;
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 2
```

The Semantic errors are correctly found because for the rem operation we want both the whole and the remainder to be integers.

Test 9:

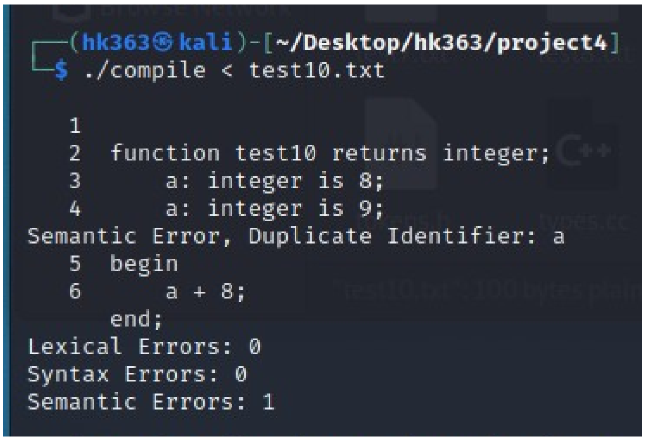


```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test9.txt

   1  -- Test of Multiple Semantic Errors
   2
   3   function test a: integer returns integer;
   4  b: integer is
   5   if a + 5 then
   6    2;
   7   else
   8   5;
   9   endif;
Semantic Error, 'If' Expression Must Be Of Boolean Type
  10
  11
  12  c: real is 9.8 - 2 + 8;
  13  d: boolean is 7 = f;
Semantic Error, Undeclared f
  14
  15
  16  begin
  17  case b is
  18  when 1 ⇒ 4.5 + c;
  19  when 2 ⇒ b;
  20
  21
  22  others ⇒ c;
Semantic Error, Narrowing Function Return Type from INT to REAL is not Allowed.
  23  endcase;
Semantic Error, Case Type Mismatch In Other Statement
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 4
```

The Semantic errors are correctly found, according to the pdf example. In addition we found one more error which is : t**he returned type of the function doesn’t match the declared return type**. This happens because if we take the **‘others’** branch at the case statement then the **type returned is real because c has real value but the function was declared to return integer type**.

Test 10:



```
┌──(hk363㉿kali)-[~/Desktop/hk363/project4]
└─$ ./compile < test10.txt

   1
   2  function test10 returns integer;
   3      a: integer is 8;
   4      a: integer is 9;
Semantic Error, Duplicate Identifier: a
   5  begin
   6      a + 8;
      end;
Lexical Errors: 0
Syntax Errors: 0
Semantic Errors: 1
```

In this test the Semantic error is correctly found, because we are trying to declare a variable while a variable with the same name already exists.