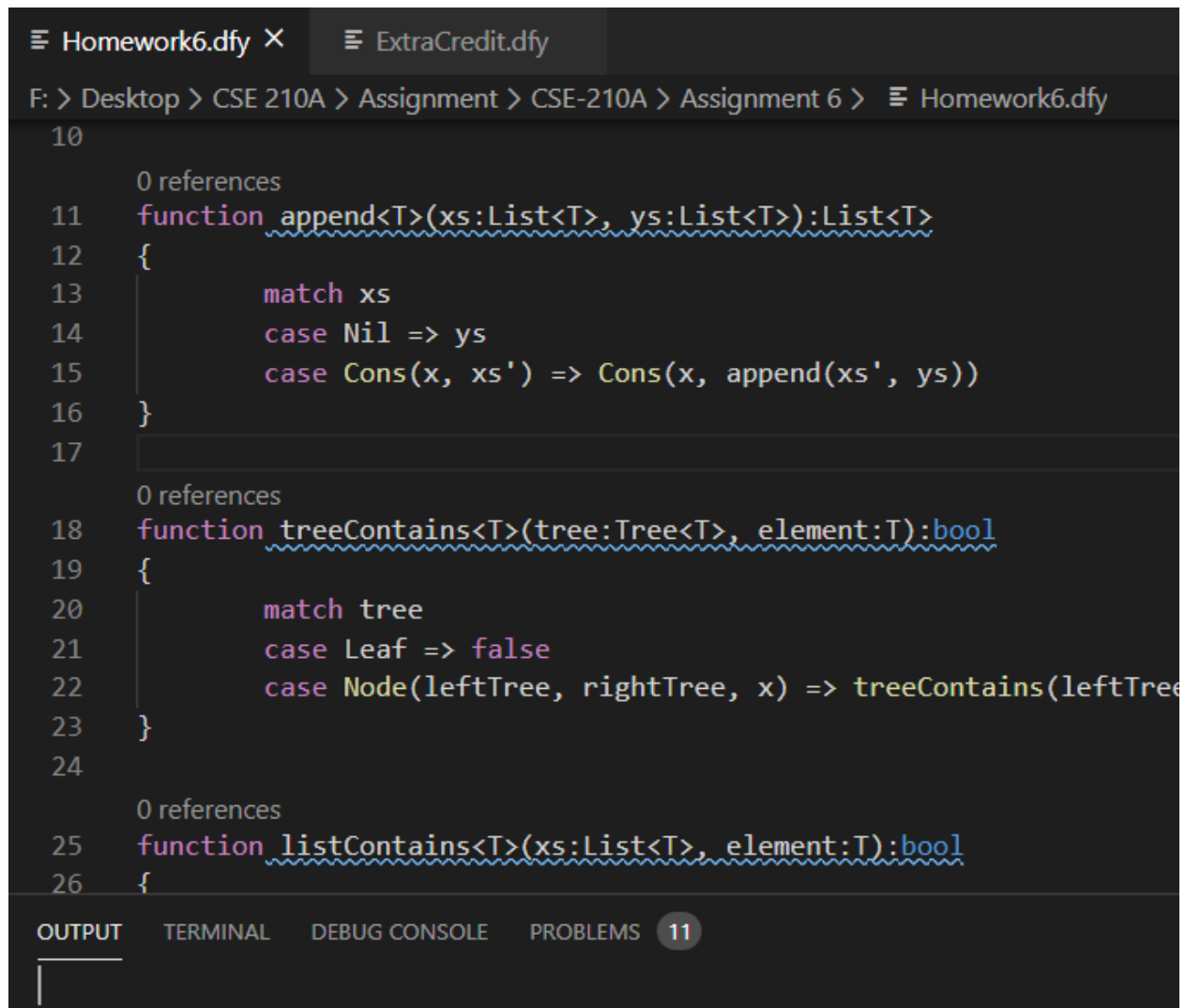


The OUTPUT section seems to be empty if there is no mistake. So I recorded the PROBLEMS section also.



```
10
    0 references
11  function append<T>(xs:List<T>, ys:List<T>):List<T>
12  {
13      match xs
14      case Nil => ys
15      case Cons(x, xs') => Cons(x, append(xs', ys))
16  }
17
    0 references
18  function treeContains<T>(tree:Tree<T>, element:T):bool
19  {
20      match tree
21      case Leaf => false
22      case Node(leftTree, rightTree, x) => treeContains(leftTree, element) || treeContains(rightTree, element)
23  }
24
    0 references
25  function listContains<T>(xs:List<T>, element:T):bool
26  {
    }
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 11

Homework6.dfy

ExtraCredit.dfy X

F:\Desktop > CSE 210A > Assignment > CSE-210A > Assignment 6 > ExtraCredit.dfy

```

54     case(Var(x), Mult(x', y')) => Mult(e1', e2')
55
56     case(Plus(x', y'), Const(b)) => if b == 0 then Const(0) else
57     case(Plus(x', y'), Var(y)) => Mult(e1', e2')
58     case(Plus(x', y'), Plus(m', n')) => Mult(e1', e2')
59     case(Plus(x', y'), Mult(m', n')) => Mult(e1', e2')
60
61     case(Mult(x', y'), Const(b)) => if b == 0 then Const(0) else
62     case(Mult(x', y'), Var(y)) => Mult(e1', e2')
63     case(Mult(x', y'), Plus(m', n')) => Mult(e1', e2')
64     case(Mult(x', y'), Mult(m', n')) => Mult(e1', e2')
65   }
66 }
67
68 //as you write optimize this will become unproved
69 //you must write proof code so that Dafny can prove this
    4 references
70 method optimizeCorrect(e:Exp, s:map<string, int>)
71 ensures eval(e,s) == eval(optimize(e), s)
72 {

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 11

ExtraCredit.dfy F:\Desktop\CSE 210A\Assignment\CSE-210A\Assignment 6 3

- (i) decreases e, store Dafny VSCode [3, 9]
- (i) decreases e Dafny VSCode [13, 9]
- (i) decreases e, s Dafny VSCode [70, 7]

Homework6.dfy F:\Desktop\CSE 210A\Assignment\CSE-210A\Assignment 6 8

- (i) decreases tree Dafny VSCode [4, 9]
- (i) decreases xs, ys Dafny VSCode [11, 9]
- (i) decreases tree Dafny VSCode [18, 9]
- (i) decreases xs Dafny VSCode [25, 9]
- (i) {induction xs, ys} Dafny VSCode [33, 6]
- (i) {induction tree} Dafny VSCode [41, 6]
- (i) decreases tree Dafny VSCode [41, 6]
- (i) == Dafny VSCode [51, 12]