Wentworth Institute of Technology

COMP3350 Programming Languages

Homework 2

1. Modify the following F# function to use pattern matching instead of if-then-else. [2]

let result =

if System.DateTime.Now.Second % 2 = 0 then

"heads"

else

"tails"

printfn "%A" result

let seconds = System.DateTime.Now.Second

let result =

match seconds with

| seconds when seconds % 2 = 0 -> "heads"

| \_ -> "tails"

printfn "%A" result

1. Briefly describe what each of the following two functions does. [3]
2. let rec incrElements list =

match list with

| head :: tail -> head + 1 :: incrElements tail

| [] -> []

printfn "Result = %A" (incrElements [4; 5; 6])

We have a recursive function that increases each element by one. It works like a loop where we start by the head and ends in the tail of the list.

1. let result = List.map ((+) 1) [4; 5; 6]

printfn "Result = %A" result

Very similar to the function above – we’re still incrementing each entry of the list by one. But not recursively. We’re making use of List.map and passing the function (+) followed by the parameter (1) to increase each entry of the given list by one.

1. You can match and decompose a tuple to its constituent elements with a *Tuple pattern*. For instance, a two-dimensional point represented as a tuple can be decomposed to its individual x- and y-coordinates with a Tuple pattern: [5]

> let point = 10, 20

let x,y = point;;

val point : int \* int = (10, 20)

val y : int = 20

val x : int = 10

In this example, the values 10 and 20 are extracted from point and bound to the x and y identifiers, respectively.

Similarly, you can use several Tuple patterns within a match expression to perform branching based upon the tupled values.

In the function locatePoint below, fill in the blanks indicating whether a particular point is located at the origin, or along an axis.

let locatePoint p =

match p with

| u (0, 0) -> printfn "%A is origin " p

| v (\_, 0) -> printfn "%A is x-axis " p

| w (0, \_) -> printfn "%A is y-axis " p

| (x, y) -> printfn "Point (%i, %i)" x y