Sault College

Name: Jangbu Sherpa

Student ID: 22054069

Professor: Nawaz Chowdhury

Course: Programming Concepts 2

Code: CSD-215

Due Date: In-class

Table of Contents

1. Create a list that displays the salaries of a company's employees. For this list, let salaries = [$75,000; $48,000; $120,000; $190,000; $300,113; $92,000; $36,000].

a. Filter through the list to find high-income salaries. For this list, salaries above $100,000 are considered high.

Program.fs

```
let salaries = [75000; 48000; 120000; 190000; 300113; 92000; 36000]


let highIncome = salaries |> List.filter (fun salary -> salary > 100000)

printfn("High income salaries: ")
highIncome |> List.iter (fun salary -> printfn "$%d" salary)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
jangbusherpa@Jangbus-MacBook-Air fsharpapp % dotnet run
High income salaries:
$120000
$190000
$300113
jangbusherpa@Jangbus-MacBook-Air fsharpapp %
```

b. Use the map function to calculate tax for all salaries based on the table provided.

Program.fs

```fsharp
let TaxCalc salary =
    match salary with
    | s when s <= 49020 -> int (float s * 0.15)
    | s when s <= 98040 -> int (float s * 0.205)
    | s when s <= 151978 -> int (float s * 0.26)
    | s when s <= 216511 -> int (float s * 0.29)
    | _ -> int (float salary * 0.33)

let salaries = [75000; 48000; 120000; 190000; 300113; 92000; 36000]

let taxes = salaries |> List.map TaxCalc

taxes |> List.iteri (fun i tax -> printfn "Salary: $%d, Tax: $%d" (List.item i salaries) tax)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh

```
jangbusherpa@Jangbus-MacBook-Air fsharpapp % dotnet run
Salary: $75000, Tax: $15374
Salary: $48000, Tax: $7200
Salary: $120000, Tax: $31200
Salary: $190000, Tax: $55099
Salary: $300113, Tax: $99037
Salary: $92000, Tax: $18860
Salary: $36000, Tax: $5400
jangbusherpa@Jangbus-MacBook-Air fsharpapp %
```

c. Filter salaries less than $49,020 and add $20,000 to these salaries using the map function.

Program.fs

```fsharp
let salaries = [75000; 48000; 120000; 190000; 300113; 92000; 36000]


let afterSalary =
    salaries
    |> List.filter (fun salary -> salary < 49020)
    |> List.map (fun salary -> salary + 20000)


afterSalary |> List.iter (fun salary -> printfn "Salaries After Addition: $%d" salary)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
jangbusherpa@Jangbus-MacBook-Air fsharpapp % dotnet run
Salaries After Addition: $68000
Salaries After Addition: $56000
jangbusherpa@Jangbus-MacBook-Air fsharpapp %
```

d. Filter salaries between $50,000 and $100,000 and sum them all using the reduce/fold function.

Program.fs

```
let salaries = [75000; 48000; 120000; 190000; 300113; 92000; 36000]
let salaryFilter = salaries |> List.filter (fun salary -> salary >= 50000 && salary <= 100000)
let sum = salaryFilter |> List.fold (fun acc salary -> acc + salary) 0
printfn "Sum of salaries between $50,000 and $100,000: $%d" sum
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh

```
jangbusherpa@Jangbus-MacBook-Air fsharpapp % dotnet run
Sum of salaries between $50,000 and $100,000: $167000
jangbusherpa@Jangbus-MacBook-Air fsharpapp %
```

2. Tail Recursion

Use tail recursion to write a program that will calculate the sum of all multiples of 3 up to a given number. Assume that we pass only a multiple of 3 as a parameter to this function. No validation is needed.

Example: If the parameter is 27, then the result of the function should be: 3+6+9+12+15+18+21+24+27 = 135.

Program.fs U

Program.fs

```fsharp
let calculate n =
    let rec loop sum multiple1 =
        if multiple1 > n then
            sum
        else
            loop (sum + multiple1) (multiple1 + 3)

    loop 0 3


let result = calculate 45
printfn "Result: %d" result

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
jangbusherpa@Jangbus-MacBook-Air fsharpapp % dotnet run
Result: 360
jangbusherpa@Jangbus-MacBook-Air fsharpapp %
```

3. Uploading Repo to github