Master of Computer and Information Science

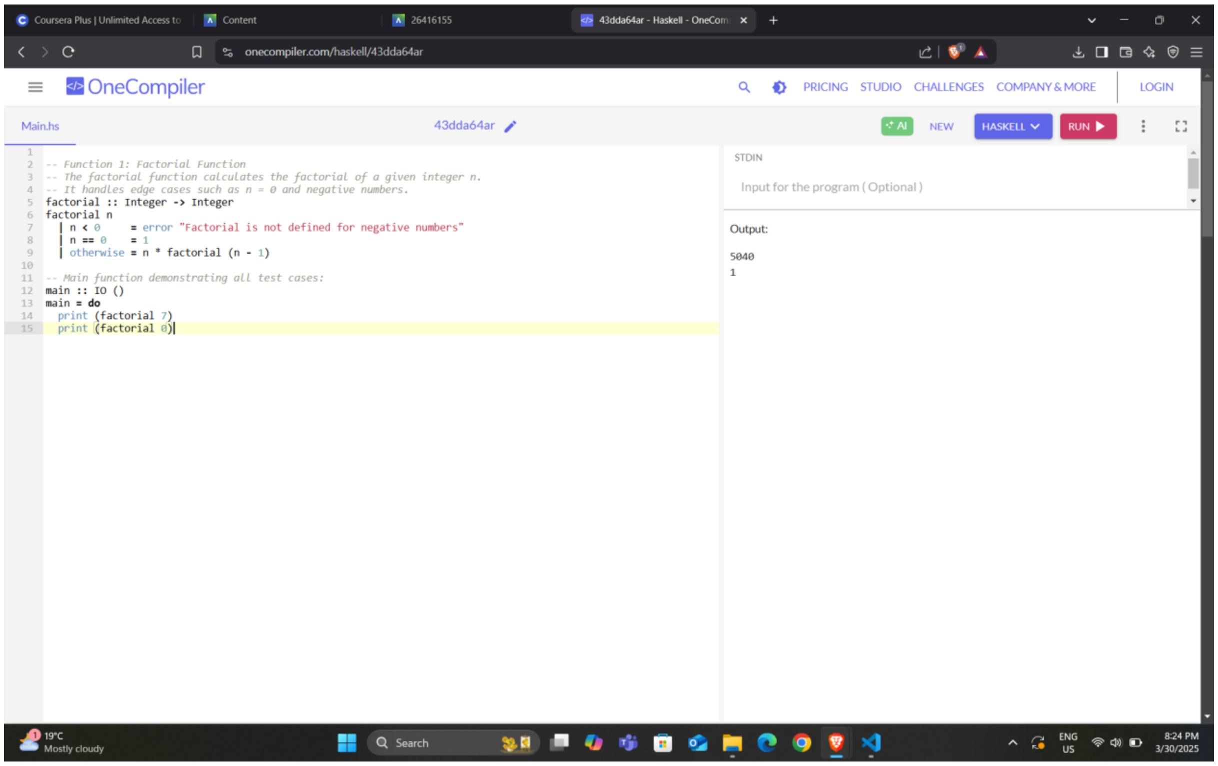
CIS 524

Haskell Programming Assignment

Name – Kathan Urmishbhai Patel

CSU ID – 2886869

1. Screenshot of Output of test case of Factorial function

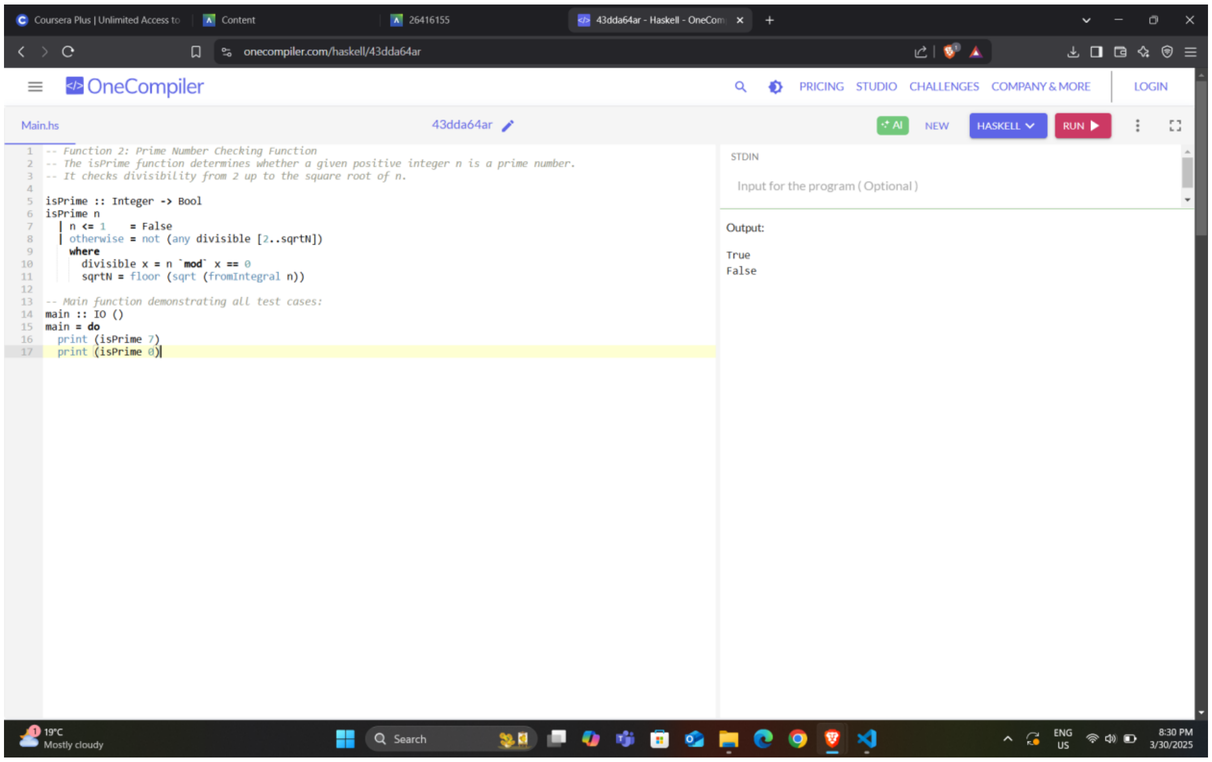


```haskell

-- Function 1: Factorial Function
-- The factorial function calculates the factorial of a given integer n.
-- It handles edge cases such as n = 0 and negative numbers.
factorial :: Integer -> Integer
factorial n
  | n < 0     = error "Factorial is not defined for negative numbers"
  | n == 0    = 1
  | otherwise = n * factorial (n - 1)

-- Main function demonstrating all test cases:
main :: IO ()
main = do
  print (factorial 7)
  print (factorial 0)
```

Output:

```
5040
1
```

2. Screenshot of Output of test case of Prime Checking function

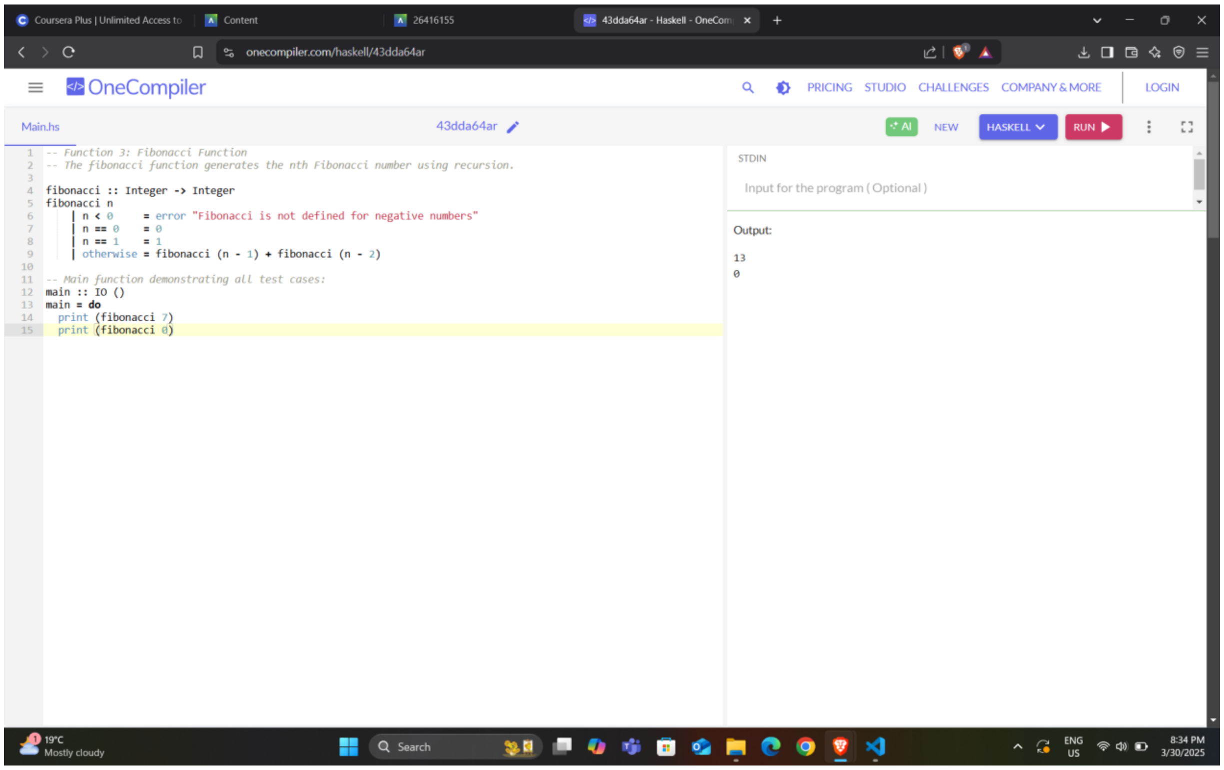


```haskell
-- Function 2: Prime Number Checking Function
-- The isPrime function determines whether a given positive integer n is a prime number.
-- It checks divisibility from 2 up to the square root of n.

isPrime :: Integer -> Bool
isPrime n
  | n <= 1    = False
  | otherwise = not (any divisible [2..sqrtN])
    where
      divisible x = n `mod` x == 0
      sqrtN = floor (sqrt (fromIntegral n))

-- Main function demonstrating all test cases:
main :: IO ()
main = do
  print (isPrime 7)
  print (isPrime 0)
```

Output:

```
True
False
```

3. Screenshot of Output of test case of Fibonacci function

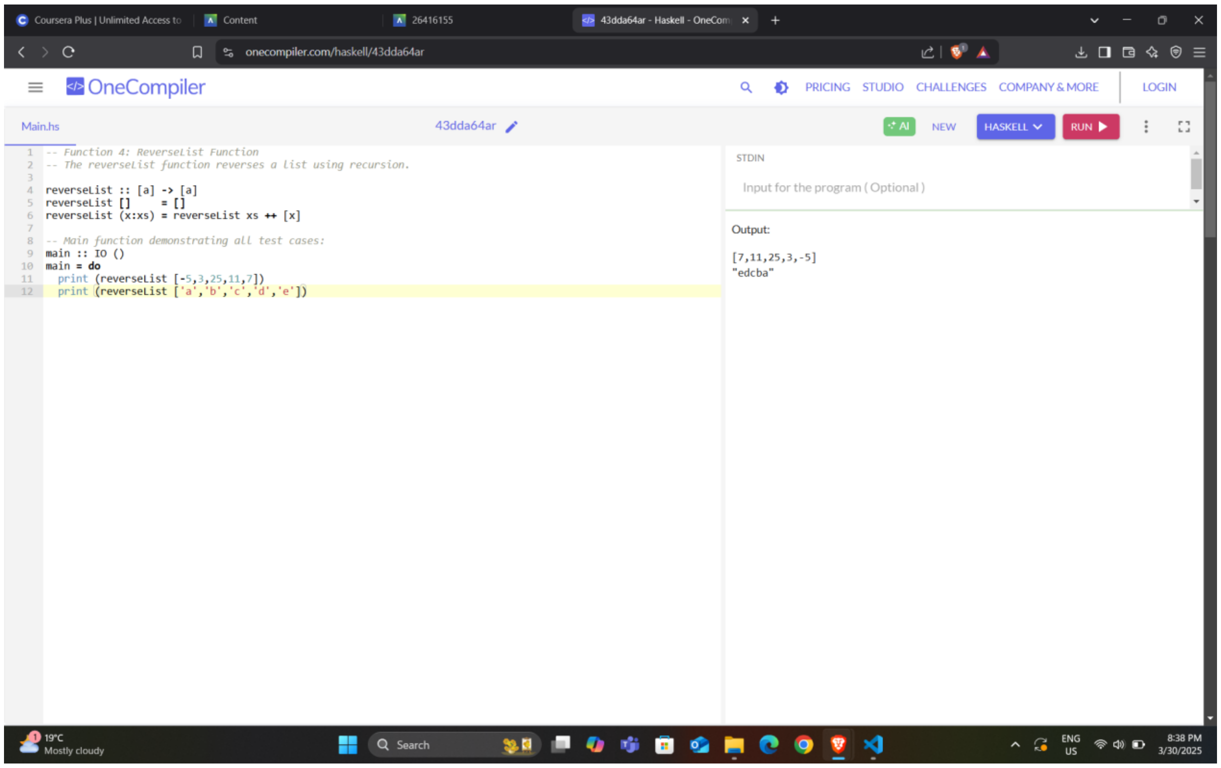


```haskell
-- Function 3: Fibonacci Function
-- The fibonacci function generates the nth Fibonacci number using recursion.

fibonacci :: Integer -> Integer
fibonacci n
    | n < 0     = error "Fibonacci is not defined for negative numbers"
    | n == 0    = 0
    | n == 1    = 1
    | otherwise = fibonacci (n - 1) + fibonacci (n - 2)

-- Main function demonstrating all test cases:
main :: IO ()
main = do
  print (fibonacci 7)
  print (fibonacci 0)
```

Output:

```
13
0
```

4. Screenshot of Output of test case of ReverseList function

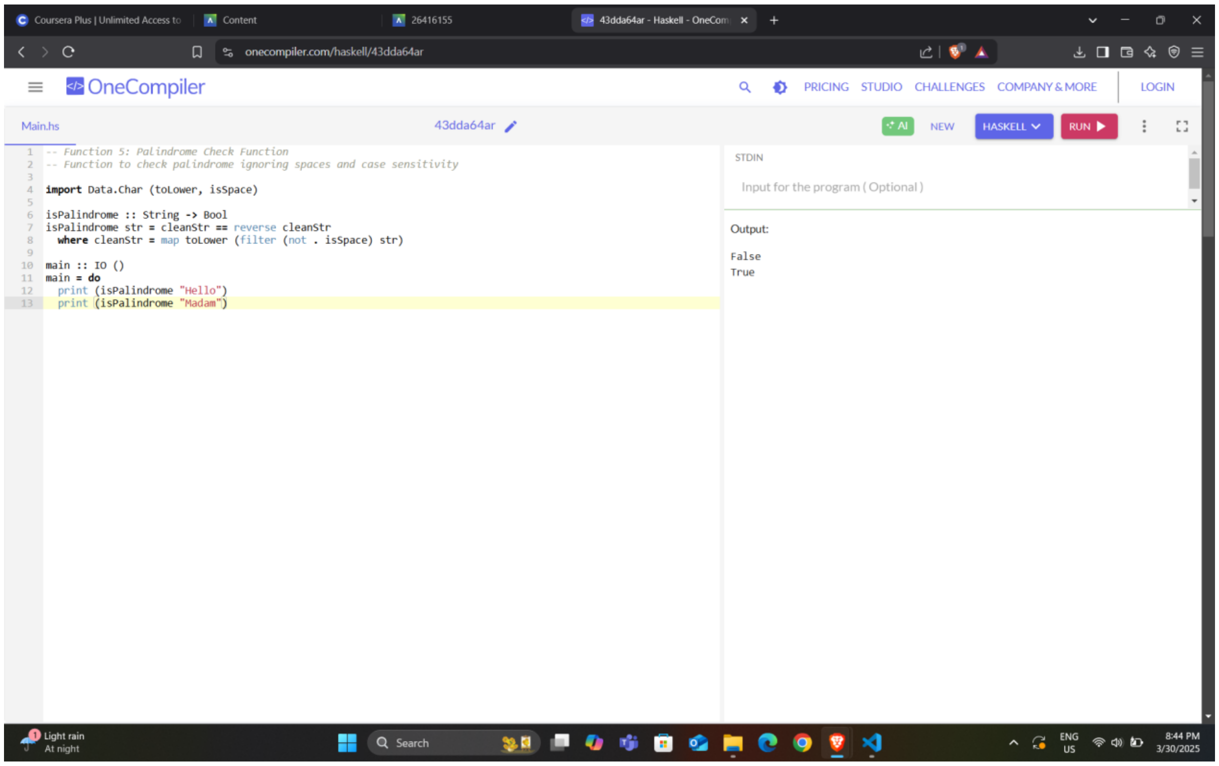


```haskell
-- Function 4: ReverseList Function
-- The reverseList function reverses a list using recursion.

reverseList :: [a] -> [a]
reverseList []     = []
reverseList (x:xs) = reverseList xs ++ [x]

-- Main function demonstrating all test cases:
main :: IO ()
main = do
  print (reverseList [-5,3,25,11,7])
  print (reverseList ['a','b','c','d','e'])
```

Output:

```
[7,11,25,3,-5]
"edcba"
```

5. Screenshot of Output of test case of Palindrome function



onecompiler.com/haskell/43dda64ar
OneCompiler
PRICING STUDIO CHALLENGES COMPANY & MORE LOGIN
Main.hs
43dda64ar
AI NEW HASKELL RUN
STDIN
Input for the program (Optional)

```haskell
-- Function 5: Palindrome Check Function
-- Function to check palindrome ignoring spaces and case sensitivity

import Data.Char (toLower, isSpace)

isPalindrome :: String -> Bool
isPalindrome str = cleanStr == reverse cleanStr
  where cleanStr = map toLower (filter (not . isSpace) str)

main :: IO ()
main = do
  print (isPalindrome "Hello")
  print (isPalindrome "Madam")
```

Output:

```
False
True
```