KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Programavimo kalbų teorija (P175B124)

Laboratorinių darbų ataskaita

Atliko:

IFF – 6/8 gr. studentas

Tadas Laurinaitis

2019 m. gegužės 20 d.

Priėmė:

Lekt. Evaldas Guogis

KAUNAS 2019

TURINYS

1. Python arba Ruby (L1) 3

1.1. Darbo užduotis 3

1.2. Programos tekstas 3

1.3. Pradiniai duomenys ir rezultatai 4

2. Scala (L2) 4

3. Haskell (L3) 4

4. Prolog (L4) 4

#### 1.1 Darbo užduotis

Pav #1 Darbo užduotis

#### 1.2 Programos tekstas

#Number class used to store a number and a number which shows a cumulative sum of each numbers divisors from 1 to number

class Number:

def \_\_init\_\_(self, number, cumulativeSum):

self.number = number

self.cumulativeSum = cumulativeSum

def get\_number(self):

return self.number

#finds sum of all viable divisors of number n

def findSumOfDivisors(n):

sum = 0

for x in range(2, int(n)):

z = n / x #temporary result of division

if z == int(z):

sum = sum + z

return sum

#finds cumulative sum of divisors for numbers 1 to Number.number

def findCumulativeSumOfDivisors(Number):

for x in range(0, Number.number + 1):

Number.cumulativeSum = Number.cumulativeSum + findSumOfDivisors(x)

print("Cumulative sum of divisors of number n: " + str(Number.number) + " is: " + str(Number.cumulativeSum))

return Number

#reads data from file into integer array

def readIntoArray(fileName):

array = []

with open('data.txt') as f:

for line in f: # read all lines

array.append(int(line))

return array

#finds results for all integers in array

def findResults(array):

numberArray = []

for x in array:

temp = Number(x, 0)

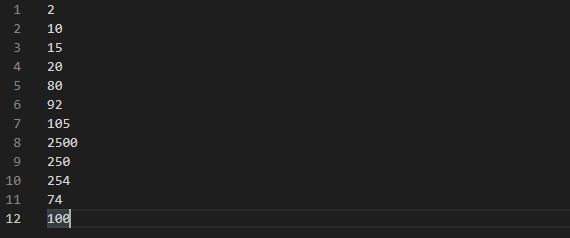
temp = findCumulativeSumOfDivisors(temp)

numberArray.append(temp)

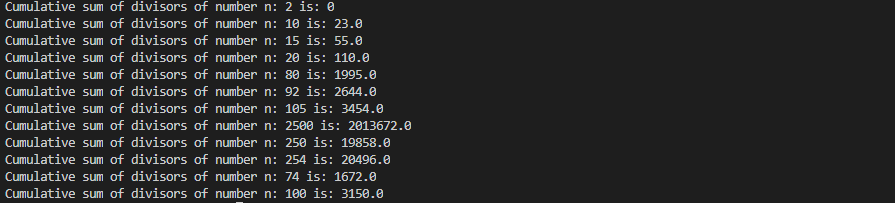
array = readIntoArray("data.txt")

findResults(array)

#### 1.3 Pradiniai duomenys ir rezultatai



Pav #2 Duomenų failas



Pav #3 Gauti rezultatai

#### Darbo užduotis

Pav #1 Darbo užduotis

#### 1.2 Programos tekstas

// Tadas Laurinaitis, IFF - 6/8, uzduoties nr. - 294, Divisors https://uva.onlinejudge.org/index.php?option=com\_onlinejudge&Itemid=8&category=4&page=show\_problem&problem=230

open System

open System.IO

let readDataFromFile file =

File.ReadAllLines(file)

let writeResultToFile file L U maxNum maxCount =

let file = File.AppendText(file)

Console.WriteLine("Between {0} and {1}, {2} has a maximum of {3} divisors", L, U, maxNum, maxCount)

file.WriteLine("Between {0} and {1}, {2} has a maximum of {3} divisors", L, U, maxNum, maxCount)

file.Close()

//let rec findDivisionsOfNumber(number : int, divisionCount : int, current : int) =

let rec findDivisionsOfNumber number divisionCount current =

if (number % current = 0 && current <= number) then

let temp1 = divisionCount + 1

let temp2 = current + 1

findDivisionsOfNumber number temp1 temp2

else if (number % current <> 0 && current <= number) then

let temp2 = current + 1

findDivisionsOfNumber number divisionCount temp2

else

let temp = divisionCount

temp

let rec findNumber L U current maxNum maxCount =

let divisionCount = findDivisionsOfNumber current 0 1

//Console.WriteLine(divisionCount)

if(divisionCount > maxCount && current <= U) then

let nextStep = current + 1

let currentMaxNum = current

let currentMaxCount = divisionCount

findNumber L U nextStep currentMaxNum divisionCount

else if (divisionCount <= maxCount && current <= U) then

let nextStep = current + 1

findNumber L U nextStep maxNum maxCount

else

let temp = maxNum

writeResultToFile "Results.txt" L U maxNum maxCount

temp

let rec doStuff current =

let numbers = readDataFromFile("D:\Tadas\KALBUTEORIJA\Fsharp\Lab3\Lab3\Data.txt")

let firstLine = numbers.[0].Split(' ')

let NN = Int32.Parse(firstLine.[0])

if(current > NN) then

printf "Job is done "

else

let strings = numbers.[current].Split(' ')

let L = Int32.Parse(strings.[0])

let U = Int32.Parse(strings.[1])

Console.WriteLine(NN)

Console.WriteLine(current)

Console.WriteLine(L)

Console.WriteLine(U)

if (U - L >= 0 && U - L <= 10000 && current <= NN) then

printf "Data looks fine "

let num = findNumber L U L 0 0

let nextStep = current + 1

doStuff nextStep

else if(U - L < 0 || U - L > 10000 && current <= NN) then

printf "The Data is incorrect "

let nextStep = current + 1

doStuff nextStep

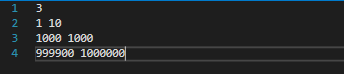
[<EntryPoint>]

let main argv =

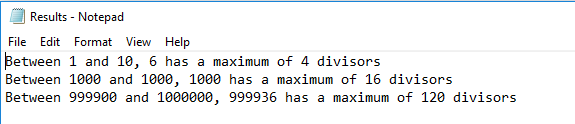
let k = doStuff 1

0

#### 1.3 Pradiniai duomenys ir rezultatai



Pav #2 Duomenų failas



Pav #3 Gauti rezultatai

#### Darbo užduotis

Darbo užduotis:

1. Surasti ar du skaičiai yra kopiriminiai
2. Surasti ar skaičius yra pirminis

#### Programos tekstas

1. % Tadas Laurinaitis IFF - 6/8 suma 39, 12 ir 13
2. % Surasti ar du skaiciai yra kopirminiai - 12 uzduotis
3. coprime(X, Y, Div, Output) :-
4. (X == Div ; Y == Div ->
5. write(" Finished ")
6. ;(X > Div, Y > Div ->
7. write("Still going "),
8. R1 is mod(X, Y),
9. R2 is mod(Y, X),
10. (R1 \== 0, R2 \== 0 ->
11. write("3333\_"),
12. write(Div),
13. R3 is mod(X, Div),
14. R4 is mod(Y, Div),
15. write(R3),
16. write(R4),
17. (( R3 \== 0 ; R4 \== 0) ->
18. write("4444"),
19. Output = 'true',
20. Z is Div + 1,
21. coprime(X, Y, Z, Output)
22. ;(R3 == 0, R4 == 0 ->
23. write("5555"),
24. Output2 = 'false',
25. coprime(100, 100, 100, Output2)
26. )
27. )
28. ;(R1 == 0 ; R2 == 0 ->
29. write("6666"),
30. Output = 'false'
31. )
32. )
33. )
34. ).
35. %Surasti ar skaicius yra pirminis - 13 uzduotis
36. isprime(1) :-
37. write("The number 1 is prime ").
38. isprime(2) :-
39. write("The number 2 is prime ").
40. isprime(X) :-
41. X > 2,
42. isprimee(X, 2, Output).
43. isprimee(X, Y, Output) :-
44. (Y < X ->
45. R1 is mod(X, Y),
46. (R1 \== 0 ->
47. Output = "true",
48. isprimee(X, Y+1, Output)
49. ; (R1 == 0 ->
50. write("The number is not a prime ")
51. )
52. )
53. ; ( Y >= X ->
54. Output = "true",
55. write("The number "),
56. write(X),
57. write(" is prime. ")
58. )
59. ).
60. /\*
61. something(X, Y) :-
62. ( X > Y ; X < Y ->
63. write("11111111"),
64. write("111111111111111111"),
65. something(X - 1,Y)
66. ; ( X = Y ->
67. write("aaa222222222"),
68. write("bbb2222222222222222222"),
69. something(X,Y+1)
70. )
71. ).
72. coprime(X, Y, Divider, Output) :-
73. R1 is mod(X,Y),
74. R2 is mod(Y,X),
75. (R1 \== 0, R2 \== 0 ->
76. R3 is mod(X, Divider),
77. R4 is mod(Y, Divider),
78. (R3 \== 0, R4 \== 0 ->
79. Output = "true",
80. coprime(X, Y, Divider, Output),
81. ; (R3 == 0 ; R4 == 0 ->
82. Output = "false"
83. )
84. )
85. ; (R1 == 0 ; R2 == 0 ->
86. Output = "false";
87. )
88. ).
89. coprime(X, Y, Div, Output) :-
90. write("iejau"),
91. ( X == Div ; Y == Div ->
92. write("as lygus esu"),
93. Output = "true",
94. coprime(X, Y, Div + 1, Output)
95. )
96. R1 is mod(X, Y),
97. R2 is mod(Y, X),
98. (R1 \== 0, R2 \== 0 ->
99. write("praslydau 1"),
100. R3 is mod(X, Div),
101. R4 is mod(Y, Div),
102. (R3 \== 0, R4 \== 0 ->
103. write("praslydau 2"),
104. Output = "true",
105. coprime(X, Y, Div+1, Output)
106. ; (R3 == 0 ; R4 == 0 ->
107. write("nepraslydau 2"),
108. Output = "false"
109. )
110. )
111. ; ( R1 == 0 ; R2 == 0 ->
112. write("nepraslydau 1"),
113. Output = "false"
114. )
115. ).
116. isprimaryy(X, Y, Output) :-
117. (Y < X ->
118. R1 is mod(X, Y),
119. (R1 \== 0 ->
120. Output = "true",
121. isprimaryy(X, Y+1, Output)
122. ; (R1 == 0 ->
123. Output = "false"
124. )
125. )
126. ; ( Y >= X ->
127. Output = "true",
128. write("The number "),
129. write(X),
130. write(" is prime. ")
131. )
132. ).
133. coprimary(X, Y) :-
134. isprimary(X, Output1),
135. isprimary(Y, Output2),
136. Output1 == "true",
137. Output2 == "true",
138. R1 is mod(X,Y),
139. R2 is mod(Y,X),
140. R1 \== 0,
141. R2 \== 0,
142. write("Numbers are coprime").
143. isprimary(1, Output) :-
144. Output = "true",
145. write("The number 1 is prime ").
146. isprimary(2, Output) :-
147. Output = "true",
148. write("The number 2 is prime ").
149. isprimary(X, Output) :-
150. X > 2,
151. isprimaryy(X, 2, Output).
152. isprimaryy(X, Y, Output) :-
153. (Y < X ->
154. R1 is mod(X, Y),
155. (R1 \== 0 ->
156. Output = "true",
157. isprimaryy(X, Y+1, Output)
158. ; (R1 == 0 ->
159. Output = "false"
160. )
161. )
162. ; ( Y >= X ->
163. Output = "true",
164. write("The number "),
165. write(X),
166. write(" is prime. ")
167. )
168. ).
169. jealous(X, Y) :-
170. loves(X, Z),
171. loves(Y, Z).
172. \*/
173. /\*\* <examples>
174. ?- loves(X, mia).
175. ?- jealous(X, Y).
176. \*/

#### 1.3 Pradiniai duomenys ir rezultatai

1užduotis:

Įvedus užklausą *coprime*(15, 18, 2, Out), gaunamas rezultatas Out -> True, o įvedus užklausą

*coprime*(15, 5, 2, Out), Out reikšmė yra false.

2užduotis:

Įvedus užklausą isprimee(3, 2, Output), gaunamas rezultatas Output -> True, o įvedus užklausą

isprimee(4, 2, Output), Output reikšmė yra false.