KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Programavimo kalbų teorija (P175B124)

Laboratorinių darbų ataskaita

Atliko:

IFF-0/3 gr. studentė

Simona Ragauskaitė

2022 m. vasario 6 d.

Priėmė:

Lekt. Evaldas Guogis

Doc. Svajūnas Sajavičius

KAUNAS 2022

TURINYS

1. Python arba Ruby (L1) 3

1.1. Darbo užduotis 3

1.2. Programos tekstas 4

1.3. Pradiniai duomenys ir rezultatai 6

# Python arba Ruby (L1)

## Text Description automatically generatedDarbo užduotis

## Programos tekstas

import math

class Data:

    def \_\_init\_\_(self, n, m):

        self.n = n

        self.m = m

class Result:

    def \_\_init\_\_(self, n, m, c):

        self.n = n

        self.m = m

        self.c = c

def readFromFile():

    n = []

    m = []

    with open("data.txt", 'r') as file:

        lines = file.readlines()

        for line in lines:

            element1, element2 = line.split(' ')

            n.append(int(element1))

            m.append(int(element2))

    return Data(n,m)

def checkReadFilesAndChangeArrays(data):

    nArray = []

    mArray = []

    arrayLength = len(data.n)

    i = 1 # index from where cycle is started

    for i in range(arrayLength):

        if data.n[i] >= 5 and data.n[i] <= 100 and data.m[i] >= 5 and data.m[i] <= 100 and data.m[i] <= data.n[i]:

            nArray.append(data.n[i])

            mArray.append(data.m[i])

        elif  data.m[i] == 0 and data.m[i] == 0:

            break

        else:

            print(Exception("Neteisingi duomenys, neatitinka reikalavimų"))

    return Data(nArray, mArray)

def calculateCombonations(nArray, mArray):

    cArray = []

    count = len(nArray)

    i = 1 # index from where cycle is started

    for i in range(count):

        cValue = (math.factorial(nArray[i]))/(math.factorial(nArray[i]-mArray[i]) \* math.factorial(mArray[i]))

        cArray.append(int(cValue))

    return Result(nArray, mArray, cArray)

def printToConsole(result):

    i = 1  # index from where cycle is started

    size = len(result.n)

    file = open("result.txt", "r+")

    file.seek(0)

    for i in range (size):

        file.write(str(result.n[i]) + " things taken " + str(result.m[i])

        + " at a time is " + str(result.c[i]) + " exactly. \n")

        file.truncate()

data = readFromFile() # read initial data from txt

updatedArray = checkReadFilesAndChangeArrays(data) # update read data list by task intervals

result = calculateCombonations(updatedArray.n, updatedArray.m) # calculate combinations

printToConsole(result) # print result to txt file

## Pradiniai duomenys ir rezultatai

1 variantas

* Pradiniai duomenys:

data.txt

100 6

20 5

18 6

0 0

* Rezultatai:

Text

Description automatically generatedresult.txt

2 variantas

* Pradiniai duomenys:

data.txt

5 5

6 5

6 6

7 5

7 6

7 7

8 5

0 0

* Rezultatai:

result.txt

A screenshot of a computer

Description automatically generated with medium confidence