МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №1

з дисципліни «Надійність комп’ютерних систем»

на тему *«*Обчислення показників надійності за даними випробувань*»*

Виконав:

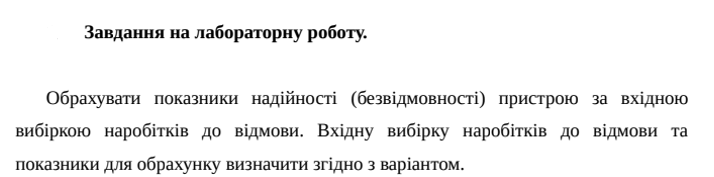
студент гр. ІО-82

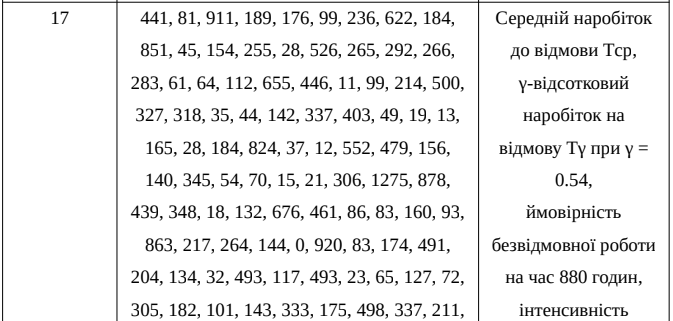
Роман О.Т.

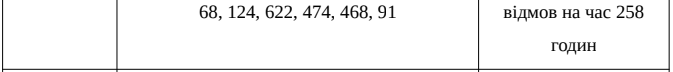
Перевірив:

Іваніщев Б.В.

Київ 2022







**Лістинг коду програми:**

***t.py:***

dataSet = [

441, 81, 911, 189, 176, 99, 236, 622, 184,

851, 45, 154, 255, 28, 526, 265, 292, 266,

283, 61, 64, 112, 655, 446, 11, 99, 214, 500,

327, 318, 35, 44, 142, 337, 403, 49, 19, 13,

165, 28, 184, 824, 37, 12, 552, 479, 156,

140, 345, 54, 70, 15, 21, 306, 1275, 878,

439, 348, 18, 132, 676, 461, 86, 83, 160, 93,

863, 217, 264, 144, 0, 920, 83, 174, 491,

204, 134, 32, 493, 117, 493, 23, 65, 127, 72,

305, 182, 101, 143, 333, 175, 498, 337, 211,

68, 124, 622, 474, 468, 91]

gamma = 0.54

timeWithoutFailures = 880

failureIntensity = 258

averageT = 0

maxTime = 0

statisticalGammaPercentFailureTime = 0

for i in range(len(dataSet)):

averageT += dataSet[i]

if dataSet[i] > maxTime:

maxTime = dataSet[i]

averageT /= len(dataSet)

print("Average time to failure:", averageT)

h = maxTime / 10

intervals = []

for i in range(10):

intervals.append([h \* i, h \* (i+1)])

statisticalDensityOfDistribution = []

for limit in range(len(intervals)):

statisticalDensityOfDistribution.append(len(list(filter(lambda i: ((i >= intervals[limit][0]) and (i <= intervals[limit][1])), dataSet)))/h/100)

sum = 0

P\_withoutFailurePerEachTime = []

for i in range(len(statisticalDensityOfDistribution)):

sum += h \* statisticalDensityOfDistribution[i]

P\_withoutFailurePerEachTime.append(1 - counter)

for i in range(len(intervals)):

if P\_withoutFailurePerEachTime[i] >= gamma > P\_withoutFailurePerEachTime[i + 1]:

statisticalGammaPercentFailureTime = (P\_withoutFailurePerEachTime[i + 1] - gamma) / (P\_withoutFailurePerEachTime[i + 1] - P\_withoutFailurePerEachTime[i])

print("The value of statistical gamma-percentage time of failure:", h - h \* statisticalGammaPercentFailureTime)

break

sum = 0

for i in range(11):

if intervals[i][1] < timeWithoutFailures:

sum += h \* statisticalDensityOfDistribution[i]

else:

P\_withoutFailureForGivenTime = 1 - sum - statisticalDensityOfDistribution[i] \* (timeWithoutFailures - intervals[i][0])

print("Probability of trouble-free operation in {} hours time:".format(timeWithoutFailures), P\_withoutFailureForGivenTime)

break

sum = 0

for i in range(11):

if intervals[i][1] < failureIntensity:

sum += statisticalDensityOfDistribution[i] \* h

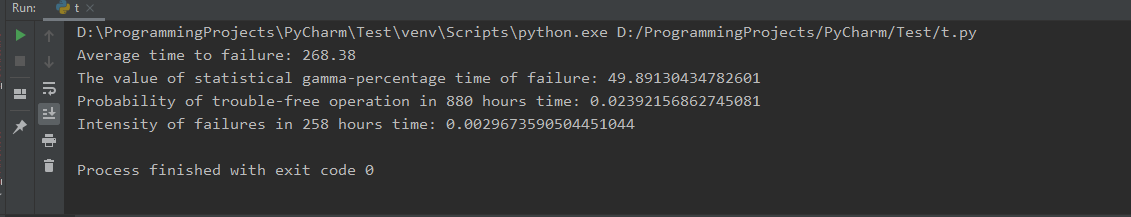
else:

print("Intensity of failures in {} hours time:".format(failureIntensity),

statisticalDensityOfDistribution[i] / (1 - sum - statisticalDensityOfDistribution[i] \* (failureIntensity - intervals[i][0])))

break

**Результати роботи програми:**



**Висновки**

В ході виконання лабораторної роботи за вхідною вибіркою наробітків до відмови були обраховані показники надійності (безвідмовності) пристрою. Результати перевірені на адекватність.