**1MPR14\_Simona\_Bļinova sb24037**

**1.uzdevums**

Programma, kas pārraksta informāciju no vienas datnes uz otro, pārveidojot burtus par uppercase.

**Kods:**

import os

import sys

# jāpievieno pareizas cēļš līdz datnei

datnes\_cels1 = 'C:/Users/Simona/Desktop/lu/programmesana un datori I/2sem/1MPR14/Simona\_Blinova\_1MPR14\_programmas\_un\_datnes/uzd1 test1.txt'

if not os.path.isfile(datnes\_cels1):

print(f'Kļuda: Datne "{datnes\_cels1}" neeksistē.')

sys.exit(1)

# jāpievieno pareizas cēļš līdz datnei

datnes\_cels2 = 'C:/Users/Simona/Desktop/lu/programmesana un datori I/2sem/1MPR14/Simona\_Blinova\_1MPR14\_programmas\_un\_datnes/uzd1 test2.txt'

if not os.path.isfile(datnes\_cels2):

print(f'Kļuda: Datne "{datnes\_cels2}" neeksistē.')

sys.exit(1)

with open('uzd1 test1.txt', 'w', encoding='utf-8') as datne:

while True:

rinda = input('Ievadiet teksta rindu --> ')

datne.write(rinda + '\n')

print('')

paz = input('Vai vēlaties ievadīt vēl rindu? (n - nē) --> ')

print('')

if paz == 'n':

break

datne.close()

with open('uzd1 test1.txt', 'r', encoding='utf-8') as datne1, \

open('uzd1 test2.txt', 'w', encoding='utf-8') as datne2:

saturs = datne1.read()

#print(saturs)

for b in saturs:

#print(b)

try:

b = b.upper()

datne2.write(b)

except:

datne2.write(b)

datne1.close()

datne2.close()

Testa piemērs(1)

Teksta datne ‘uzd1 test1.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Teksta datne ‘uzd1 test2.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A screen shot of a computer

AI-generated content may be incorrect.

Testa piemērs(2)

Teksta datne ‘uzd1 test1.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Teksta datne ‘uzd1 test2.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A screenshot of a computer screen

AI-generated content may be incorrect.

Testa piemērs(3)

Teksta datne ‘uzd1 test1.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Teksta datne ‘uzd1 test2.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:



**2.uzdevums**

Programma, kas teksta datnē saņem katra burta biežumu.

**Kods:**

import os

import sys

# jāpievieno pareizas cēļš līdz datnei

datnes\_cels = 'C:/Users/Simona/Desktop/lu/programmesana un datori I/2sem/1MPR14/Simona\_Blinova\_1MPR14\_programmas\_un\_datnes/uzd2 test1.txt'

if not os.path.isfile(datnes\_cels):

print(f'Kļuda: Datne "{datnes\_cels}" neeksistē.')

sys.exit(1)

with open('uzd2 test1.txt', 'w', encoding='utf-8') as datne:

while True:

rinda = input('Ievadiet teksta rindu --> ')

datne.write(rinda + '\n')

print('')

paz = input('Vai vēlaties ievadīt vēl rindu? (n - nē) --> ')

print('')

if paz == 'n':

break

datne.close()

vardnica = {}

with open('uzd2 test1.txt', 'r', encoding='utf-8') as datne:

simbols = datne.read(1)

while simbols != '':

while simbols != '\n' and simbols != '':

simbols = simbols.upper()

if 64 < ord(simbols) < 91:

if simbols in vardnica:

vardnica[simbols] += 1

else:

vardnica[simbols] = 1

simbols = datne.read(1)

if simbols == '\n':

simbols = datne.read(1)

#print(vardnica)

burti = list(vardnica.keys())

#print(burti)

skaiti = list(vardnica.values())

#print(skaiti)

garums = len(skaiti)

atkartojumi = garums - 1

pazime = True

while pazime:

pazime = False

for j in range(0, atkartojumi):

if skaiti[j] < skaiti[j+1]:

pazime = True

b = skaiti[j]

skaiti[j] = skaiti[j+1]

skaiti[j+1] = b

b1 = burti[j]

burti[j] = burti[j+1]

burti[j+1] = b1

atkartojumi -= 1

#print(burti)

#print(skaiti)

for i in range(len(burti)):

print(burti[i], '->', skaiti[i])

Testa piemērs(1)

Teksta datne ‘uzd2 test1.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A computer screen shot of white text

AI-generated content may be incorrect.

Testa piemērs(2)

Teksta datne ‘uzd2 test1.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A screen shot of a computer

AI-generated content may be incorrect.

Testa piemērs(3)

Teksta datne ‘uzd2 test1.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A screenshot of a computer program

AI-generated content may be incorrect.

**3.uzdevums**

Programma, kas veic teksta šifrēšanu un atšifrēšanu ar Cēzara algoritmu.

**Kods:**

teksts = input('Ievadiet tekstu --> ')

darb = input('Ko vēlaties darīt? (c - šifrēt, d - atšifrēt) --> ')

solis = int(input('Ievadiet šifrēšanas soli --> '))

teksts = teksts.upper()

simboli = []

for i in teksts:

simboli.append(i)

#print(simboli)

if darb == 'c':

solis = -solis

for i in range(len(simboli)):

#print(i)

burts = simboli[i]

#print(burts)

if 64 < ord(burts) < 91:

jaun\_burts = ord(burts) + solis

if jaun\_burts <= 64:

plus\_solis = 65 - jaun\_burts

jaun\_burts = 91 - plus\_solis

if jaun\_burts >= 91:

plus\_solis = jaun\_burts - 90

jaun\_burts = 64 + plus\_solis

simboli[i] = chr(jaun\_burts)

#print(simboli)

jaun\_teksts = ''

for elem in simboli:

jaun\_teksts += elem

print(jaun\_teksts)

Testa piemērs(1)

A screenshot of a computer screen

AI-generated content may be incorrect.

Testa piemērs(2)

A screenshot of a computer

AI-generated content may be incorrect.

Testa piemērs(3)

A screen shot of a computer code

AI-generated content may be incorrect.

**PU1**

Programma, kas teksta datnē saņem katra latviešu burta biežumu.

**Kods:**

import os

import sys

def ievietosanas\_metode(a, b):

garums = len(a)

for i in range(1, garums):

x = a[i]

xb = b[i]

j = i

while j > 0 and a[j-1] < x:

a[j] = a[j-1]

b[j] = b[j-1]

j -= 1

a[j] = x

b[j] = xb

return a, b

datnes\_cels = 'C:/Users/Simona/Desktop/lu/programmesana un datori I/2sem/1MPR14/Simona\_Blinova\_1MPR14\_programmas\_un\_datnes/PU1 test.txt'

if not os.path.isfile(datnes\_cels):

print(f'Kļuda: Datne "{datnes\_cels}" neeksistē.')

sys.exit(1)

with open('PU1 test.txt', 'w', encoding='utf-8') as datne:

while True:

rinda = input('Ievadiet teksta rindu --> ')

datne.write(rinda + '\n')

print('')

paz = input('Vai vēlaties ievadīt vēl rindu? (n - nē) --> ')

print('')

if paz == 'n':

break

datne.close()

vardnica = {}

lat\_burti = [256, 268, 274, 286, 298, 310, 315, 325, 352, 362, 381]

with open('PU1 test.txt', 'r', encoding='utf-8') as datne:

simbols = datne.read(1)

while simbols != '':

while simbols != '\n' and simbols != '':

simbols = simbols.upper()

if 64 < ord(simbols) < 81 or 81 < ord(simbols) < 87 or ord(simbols) == 90 or ord(simbols) in lat\_burti:

if simbols in vardnica:

vardnica[simbols] += 1

else:

vardnica[simbols] = 1

simbols = datne.read(1)

if simbols == '\n':

simbols = datne.read(1)

#print(vardnica)

burti = list(vardnica.keys())

#print(burti)

skaiti = list(vardnica.values())

#print(skaiti)

#print(burti)

#print(skaiti)

skaiti, burti = ievietosanas\_metode(skaiti, burti)

#print(burti)

#print(skaiti)

for i in range(len(burti)):

print(burti[i], '->', skaiti[i])

Testa piemērs(1)

Teksta datne ‘PU1 test.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Testa piemērs(2)

Teksta datne ‘PU1 test.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A screen shot of a computer code

AI-generated content may be incorrect.

Testa piemērs(3)

Teksta datne ‘PU1 test.txt’:

A screenshot of a computer

AI-generated content may be incorrect.

Terminālis:

A screenshot of a computer screen

AI-generated content may be incorrect.

**PU2**

Programma, kas veic teksta šifrēšanu un atšifrēšanu ar Cēzara šifru.

**Kods:**

lat\_burti = ['Ā', 'Č', 'Ē', 'Ģ', 'Ī', 'Ķ', 'Ļ', 'Ņ', 'Š', 'Ū', 'Ž']

alfabets = []

for i in range(65, 81):

alfabets.append(chr(i))

for i in range(82, 87):

alfabets.append(chr(i))

alfabets.append(chr(90))

i = 0

burts = alfabets[i]

while burts != 'Ž':

match burts:

case 'A':

alfabets.insert(i+1, 'Ā')

case 'C':

alfabets.insert(i+1, 'Č')

case 'E':

alfabets.insert(i+1, 'Ē')

case 'G':

alfabets.insert(i+1, 'Ģ')

case 'I':

alfabets.insert(i+1, 'Ī')

case 'K':

alfabets.insert(i+1, 'Ķ')

case 'L':

alfabets.insert(i+1, 'Ļ')

case 'N':

alfabets.insert(i+1, 'Ņ')

case 'S':

alfabets.insert(i+1, 'Š')

case 'U':

alfabets.insert(i+1, 'Ū')

case 'Z':

alfabets.insert(i+1, 'Ž')

i += 1

burts = alfabets[i]

#print(alfabets)

teksts = input('Ievadiet tekstu --> ')

darb = input('Ko vēlaties darīt? (c - šifrēt, d - atšifrēt) --> ')

solis = int(input('Ievadiet šifrēšanas soli --> '))

teksts = teksts.upper()

simboli = []

for i in teksts:

simboli.append(i)

if darb == 'c':

solis = -solis

for i in range(len(simboli)):

#print(i)

burts = simboli[i]

vieta = alfabets.index(burts)

#print(vieta)

if -1 < vieta < len(alfabets):

jaun\_vieta = vieta + solis

#print(jaun\_vieta)

if jaun\_vieta <= -1:

jaun\_vieta = len(alfabets) + jaun\_vieta

#print(jaun\_vieta)

if jaun\_vieta >= len(alfabets):

plus\_solis = jaun\_vieta - len(alfabets)

#print(plus\_solis)

jaun\_vieta = 0 + plus\_solis

simboli[i] = alfabets[jaun\_vieta]

#print(simboli)

jaun\_teksts = ''

for elem in simboli:

jaun\_teksts += elem

print(jaun\_teksts)

Testa piemērs(1)

A screenshot of a computer

AI-generated content may be incorrect.

Testa piemērs(2)

A screenshot of a computer screen

AI-generated content may be incorrect.

Testa piemērs(3)

A screenshot of a computer

AI-generated content may be incorrect.