

### Task 2.3: Playfair cracker test

**english\_text:** The sun rose slowly, casting a warm glow over the sleepy town.  
Sarah walked along the beach, feeling the soft sand between her toes.  
The old man sat on the park bench, feeding breadcrumbs to the pigeons.  
The aroma of freshly baked bread wafted through the air, making everyone's mouths water.  
Emily gazed at the night sky, mesmerized by the twinkling stars.  
The children giggled with delight as they chased colorful butterflies in the garden.  
The majestic mountains stood tall, their peaks covered in a blanket of snow.  
The sweet scent of blooming flowers filled the garden, attracting bees and butterflies.  
The sound of laughter echoed through the room as friends shared jokes and stories.  
Mark carefully placed each piece of the puzzle, determined to complete it before dinner.  
The rain poured down in torrents, drenching everyone caught outside without an umbrella.  
The orchestra played a beautiful melody, filling the concert hall with music.  
The little puppy wagged its tail, excitedly welcoming its owner home.  
Sarah's heart raced as she stood on the edge of the cliff, ready to take the leap.  
The cozy cabin nestled in the woods, offering a peaceful retreat from the hustle and bustle of city life.

**german\_text:** Die Sonne stieg langsam auf und warf einen warmen Schein über die verschlafene Stadt.  
Sarah ging am Strand entlang und spürte den weichen Sand zwischen ihren Zehen.  
Der alte Mann saß auf der Parkbank und fütterte die Tauben mit Brotkrumen.  
Der Duft von frisch gebackenem Brot zog durch die Luft und ließ jedem das Wasser im Mund zusammenlaufen.  
Emily starrte den Nachthimmel an und war fasziniert von den funkelnden Sternen.  
Die Kinder kicherten vor Freude, als sie bunte Schmetterlinge im Garten jagten.  
Die majestätischen Berge standen hoch und ihre Gipfel waren mit einer Schneedecke bedeckt.  
Der süße Duft blühender Blumen erfüllte den Garten und lockte Bienen und Schmetterlinge an.  
Das Geräusch von Gelächter hallte durch den Raum, als Freunde Witze erzählten und Geschichten teilten.  
Mark setzte jedes Puzzleteil sorgfältig ein und war entschlossen, es vor dem Abendessen fertigzustellen.  
Der Regen prasselte in Strömen herab und durchnässte jeden, der ohne Regenschirm draußen erwischt wurde.  
Das Orchester spielte eine wunderschöne Melodie und erfüllte den Konzertsaal mit Musik.  
Der kleine Welpen wedelte aufgeregt mit dem Schwanz und hieß seinen Besitzer freudig willkommen.  
Sarahs Herz raste, als sie am Rande der Klippe stand und bereit war, den Sprung zu wagen.  
Die gemütliche Hütte schmiegte sich in den Wald und bot eine friedliche Zuflucht vor dem Trubel des Stadtlebens.

**lithuanian\_text:** Saulė lėtai kildavo, paleisdama šiltą šviesą per mieguistą miestelį.  
Sara pasivaikščiojo ant paplūdimio, jautei švelnų smėlį tarp savo pirštų.  
Senelis sėdėjo parko suoloje, maitindamas balandžius trupiniais.  
Šviežiai kepto duonos kvapas sklido oru, vandenys kiekvieną burną.  
Emilija žiūrėjo į nakties dangų, žavintis mirksinčiomis žvaigždėmis.  
Vaikai džiūgavo džiaugsmingai, gaudami spalvingus drugelius sode.  
Didingi kalnai kildavo aukštai, jų viršūnės apsiklojo sniego užklotu.  
Gydančių gėlių saldus kvapas pripildė sodą, pritraukdamas bitės ir drugelius.  
Juoko garsai skleidėsi per kambarį, kai draugai dalijosi juokais ir pasakojimais.  
Markas atidžiai sudėliojo kiekvieną delionės dalį, nusiteikęs ją baigti prieš vakarienę.  
Lietus lijo lyg iš kibiro, permokdamas visus, kas buvo lauke be skėčio.  
Orkestras grojo gražią melodiją, užpildydamas koncertų salę muzika.  
Mažylis šuniukas pešė uodegą, su džiaugsmu pasitikdamas savo šeimnininką namuose.  
Saros širdis skubėjo, ji stovėjo ant skardos krašto, pasiruošusi šuoliui.  
Jauki trobelė glūdėjo miške, siūlydama ramų prieglobstį nuo miesto šurmilio.

Code:

```

import copy
import random
import math
import re

with open("english_text", "r", encoding="utf8") as f:
    english_text = f.read()

with open("german_text", "r", encoding="utf8") as f:
    german_text = f.read()

with open("lithuanian_text", "r", encoding="utf8") as f:
    lithuanian_text = f.read()

# List of texts
texts = [english_text, german_text, lithuanian_text]

# remove all non-lower case elements of the standard latin alphabet
def lclear(text):
    return re.sub(r'[^a-z]', '', text.lower()) # we use regular expressions

def Ngrams(text, n): # logarithmic fitness
    digs = dict()
    for i in range(len(text) - n):
        d = text[i:i + n]
        if d in digs.keys():
            digs[d] += 1
        else:
            digs[d] = 1

    tot = sum([digs[x] for x in digs.keys()])
    for k in digs.keys():
        digs[k] /= tot * 1.0
        digs[k] = math.log(digs[k], 10)
    flo = math.log(0.01 / tot, 10) # base score
    return digs, list(reversed(sorted([[digs[v], v] for v in digs.keys()]])), flo)

def Fitness(text, kgrams, k, f): # compute fitness of text with respect to kgrams dictionary
    score = 0
    for i in range(len(text) - k):
        d = text[i:i + k]
        if d in kgrams.keys():
            score += kgrams[d]
        else:
            score += f
    return score

def fit(text):
    return Fitness(lclear(text), n4, 4, f)

def indexof(letter, matrix):
    for i in range(5):
        try:
            index = matrix[i].index(letter)
            return (i, index)
        except:
            continue

```

```

def create_matrix(key):
    key = key.upper()
    matrix = [[0 for i in range(5)] for j in range(5)]
    letters_added = []
    row = 0
    col = 0
    # add the key to the matrix
    for letter in key:
        if letter not in letters_added:
            matrix[row][col] = letter
            letters_added.append(letter)
        else:
            continue
        if (col == 4):
            col = 0
            row += 1
        else:
            col += 1
    # Add the rest of the alphabet to the matrix
    # A=65 ... Z=90
    for letter in range(65, 91):
        if letter == 74: # I/J are in the same position
            continue
        if chr(letter) not in letters_added: # Do not add repeated letters
            letters_added.append(chr(letter))

    # print (len(letters_added), letters_added)
    index = 0
    for i in range(5):
        for j in range(5):
            matrix[i][j] = letters_added[index]
            index += 1
    return matrix

def separate_same_letters(message):
    index = 0
    while (index < len(message)):
        l1 = message[index]
        if index == len(message) - 1:
            message = message + 'X'
            index += 2
            continue
        l2 = message[index + 1]
        if l1 == l2:
            message = message[:index + 1] + "X" + message[index + 1:]
            index += 2
    return message

def playfairmat(key, message, encrypt=True):
    inc = 1
    if encrypt == False:
        inc = -1
    matrix = key
    message = message.upper()
    message = message.replace(' ', '')
    message = separate_same_letters(message)
    cipher_text = ""

```

```

for (l1, l2) in zip(message[0::2], message[1::2]):
    row1, col1 = indexOf(l1, matrix)
    row2, col2 = indexOf(l2, matrix)
    if row1 == row2: # Rule 2, the letters are in the same row
        cipher_text += matrix[row1][(col1 + inc) % 5] + matrix[row2][(col2 + inc) % 5]
    elif col1 == col2: # Rule 3, the letters are in the same column
        cipher_text += matrix[(row1 + inc) % 5][col1] + matrix[(row2 + inc) % 5][col2]
    else: # Rule 4, the letters are in a different row and column
        cipher_text += matrix[row1][col2] + matrix[row2][col1]
return cipher_text

```

```

def swap(m1):
    m = copy.deepcopy(m1)
    x = random.randint(0, 4)
    y = random.randint(0, 4)
    u = random.randint(0, 4)
    v = random.randint(0, 4)
    a = m[x][y]
    b = m[u][v]
    m[u][v] = a
    m[x][y] = b
    return m

```

```

def modify_key(m):
    return swap(m)

```

```

def playcrack(ciph, key):
    temp = 200
    step = 2
    count = 10000
    parent = key
    maxscore = fit(playfairmat(parent, ciph, False))
    bestscore = maxscore
    bestkey = parent

```

```

for tt in range(temp, 0, -step):
    t = tt / 10.0
    for _ in range(count, 0, -1):
        child = modify_key(parent)
        c = playfairmat(child, ciph, False)
        p = playfairmat(parent, ciph, False)
        score = fit(c)
        df = score - maxscore
        if df > 0:
            parent = child
            maxscore = score
        else:
            prob = math.exp(df / t)
            r = random.random()
            if prob > r:
                parent = child
                maxscore = score
        if maxscore > bestscore:
            bestscore = maxscore
            bestkey = parent
            print(bestscore)
    return bestscore, bestkey

```

```

for text in texts:
    clear_text = lclear(text)
    n4,l4,f = Ngrams(clear_text.replace('j','i'), 4)
    parent=create_matrix('ABCDEFGHIJKLMNOPQRSTUVWXYZ'.lower())
    key=create_matrix('SECRET'.lower())
    cc=playfairmat(key, clear_text[:400].replace('j','i'))
    b=-200000
    bk=parent
    while True:
        b, bk = playcrack(cc, bk)
        print(b)
        print(bk)
        print(playfairmat(bk, cc, False))

print("DONE!")

```

Output:

```

-2011.5227581170889
-2008.9206981257607
-2004.3186381344326
-2000.2217281214248
-1997.8237881127527
-1996.9206981257607
-1996.1248181084165
-1989.9206981257605
-1983.8415168797133
-1982.762335633665
-1982.2852143789453
-1981.5685156076488
-1980.6654256206568
-1978.6654256206568
-1978.1705755989767
-1960.1705755989765
-1956.6476968536963
-1956.0176081387685
-1953.8134881561123
-1951.8134881561123
-1949.8134881561123
-1945.8134881561123
-1943.8134881561123
-1943.6373968970565
-1943.160275642337
-1942.6831543876172
-1921.8134881561116
-1912.9103981691194
-1910.6093681734556
-1910.1322469187357
-1907.831216923072
-1906.7343069100639
-1905.336366901392
-1891.9561556596802

```

-1891.7800644006247  
-1890.8592456466724  
-1883.9384268927201  
-1877.8134881561116  
-1875.2114281647835  
-1868.6093681734553  
-1865.8134881561116  
-1844.1322469187357  
-1838.1322469187357  
-1788.3260669447511  
-1782.326066944751  
-1736.7697644439831  
-1736.7697644439831

[[ 'N', 'U', 'Q', 'P', 'O'], [ 'S', 'T', 'R', 'C', 'E'], [ 'H', 'M', 'L', 'K', 'T'], [ 'A', 'G', 'F', 'D', 'B'], [ 'V', 'Z', 'Y', 'X', 'W']]

THRCOPROCRSLBOYQCACEINBDWARMGLBOOVRSTHRCLEEPYTBOAHARSVWAHME  
DALQPTMHHEIACHFEXELINTMHESOFTHSNDEITWEXENHESEBICEHEOLDMSAHSTO  
NTHPARKEINCHFXEDINBFSRDGTSGZBSTOTHEPIGIBAHTHEAROMAOFRLRCKHYB  
AKEDBREADWDAERDTHRQOGHTHEAIRMAMLNGEVRSYONESMQOTHSWATRSEMLH  
YGAZEDATTHENIGHTSKYMRMERIZEGFYTHETWINMHINGSTASCTHRTKLLDSRNGI  
GGLEDOWTHDEHLGHTACEHEYCVSCRCKOLORFULBGMERLRHLRCINTHEGARDENT  
HEMAWICEICMOOPTAINSXCXODTALXLTHEIRPEAKCTOVR

-1649.1288856652795

-1649.1288856652795

[[ 'B', 'A', 'F', 'G', 'D'], [ 'T', 'H', 'L', 'M', 'K'], [ 'W', 'V', 'Y', 'Z', 'X'], [ 'O', 'N', 'Q', 'U', 'P'], [ 'E', 'S', 'R', 'T', 'C']]

THTCOPROCTSLIOYQCACEINBDWARMGLIOOVTRTHTCLEEPYTIOHSARAVWALHEDA  
LUPGTHEBEACHFEXELINGTHESOFTSANDBETWEXENHEREIECEHEOLDMAHSATONT  
HEPARKBENCHFEXEDINBGRTDASRMZBSTOTHEPIGEIHSTHEAROMAOFRTCKLYBAK  
EDBREADWDFETDTHRUOGHTHEAIRMAMNGEVTRYONESMUOTHSWATTREMMLYG  
AZEDATTHENIGHTSKYMTCMERIZEAGYTHETWINHLINGSTARCTHTSKMLDRTNGIGG  
LEDOWTHDELMGHTACEHEYCVACTDCOLORFULBMTETRFLMTCINTHEGARDENTHE  
MAWECEICMOOPTAINSXCXODTALXLTHEIRPEAKCSOVTR

-1645.8455844365758

-1628.6237356869594

-1528.641464453918

-1528.641464453918

[[ 'X', 'V', 'Z', 'Y', 'W'], [ 'C', 'S', 'T', 'R', 'E'], [ 'D', 'A', 'G', 'F', 'B'], [ 'K', 'H', 'M', 'L', 'T'], [ 'P', 'N', 'U', 'Q', 'O']]

THERPQRORESLOWLYCARCINDFWARMGLOWOVESTHERLEEPYTOWNSSARAHWAHM  
EDALOQGTHEBEACHFEXELINGTHESOFTSANDBETWEXENHESCOERCHEOLDMANSA  
TONTHEPARKBENCHFEXEDINDBSEFGTSUMBSTOTHEPIGEONSTHEAROMAOFRRERL  
HYBAKEDBREADWFACEDTHROPGHTHEAIRMAMINGEVESYONESMOPTHSWATESEMI  
HYGAZEDATTHENIGHTSKYMERMERIZEGBYTHETWINMHINGSTASRTHETLILDSENGI  
GGLEDWITHDEHIGHTARCHEYCHAREDCOLORFULBUTCERFHIERINTHEGARDENTHE  
MAIERCICMOPQTAINSXRXCXODTALXLTHEIRPEAKRTOVES

-1407.2386195204908

-1407.2386195204908

[[ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'V', 'W', 'X', 'Y', 'Z'], [ 'H', 'T', 'K', 'L', 'M'], [ 'N', 'O', 'P', 'Q', 'U']]

THESUNROESLBOLQCASTINGAWARMGLBOOVERTHESLEEPYTBOASARVHWALKED  
ALONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHRTBESTHEOLDMVASVTO

NTHEPARKWENCHFEXEDINGBREADCRGMBSTOTHEPIGEBASTHEAROMAOFYRESHLY  
YBAKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMIL  
YGAZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIG  
GLEDOITHDELIGHTASTHEYCHVSEXCOLORFULBGTTERYLIESINTHEGARDENTHEMA  
IESTICMOUNTAINSXSXODTALXLTHEIRPEAKSCOVER

-1407.2386195204908

[[ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'V', 'W', 'X', 'Y', 'Z'], [ 'H', 'T', 'K', 'L', 'M'], [ 'N', 'O', 'P', 'Q', 'U'] ]

THESUNROSESLBOLQCASTINGAWARMGLBOOVERTHESLEEPYTBOASARVHWALKED  
ALONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTBESTHEOLDMVASVTO  
NTHEPARKWENCHFEXEDINGBREADCRGMBSTOTHEPIGEBASTHEAROMAOFYRESHLY  
YBAKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMIL  
YGAZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIG  
GLEDOITHDELIGHTASTHEYCHVSEXCOLORFULBGTTERYLIESINTHEGARDENTHEMA  
IESTICMOUNTAINSXSXODTALXLTHEIRPEAKSCOVER

-1407.2386195204908

[[ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'V', 'W', 'X', 'Y', 'Z'], [ 'H', 'T', 'K', 'L', 'M'], [ 'N', 'O', 'P', 'Q', 'U'] ]

THESUNROSESLBOLQCASTINGAWARMGLBOOVERTHESLEEPYTBOASARVHWALKED  
ALONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTBESTHEOLDMVASVTO  
NTHEPARKWENCHFEXEDINGBREADCRGMBSTOTHEPIGEBASTHEAROMAOFYRESHLY  
YBAKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMIL  
YGAZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIG  
GLEDOITHDELIGHTASTHEYCHVSEXCOLORFULBGTTERYLIESINTHEGARDENTHEMA  
IESTICMOUNTAINSXSXODTALXLTHEIRPEAKSCOVER

-1407.2386195204908

[[ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'V', 'W', 'X', 'Y', 'Z'], [ 'H', 'T', 'K', 'L', 'M'], [ 'N', 'O', 'P', 'Q', 'U'] ]

THESUNROSESLBOLQCASTINGAWARMGLBOOVERTHESLEEPYTBOASARVHWALKED  
ALONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTBESTHEOLDMVASVTO  
NTHEPARKWENCHFEXEDINGBREADCRGMBSTOTHEPIGEBASTHEAROMAOFYRESHLY  
YBAKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMIL  
YGAZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIG  
GLEDOITHDELIGHTASTHEYCHVSEXCOLORFULBGTTERYLIESINTHEGARDENTHEMA  
IESTICMOUNTAINSXSXODTALXLTHEIRPEAKSCOVER

-1407.2386195204908

[[ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'V', 'W', 'X', 'Y', 'Z'], [ 'H', 'T', 'K', 'L', 'M'], [ 'N', 'O', 'P', 'Q', 'U'] ]

THESUNROSESLBOLQCASTINGAWARMGLBOOVERTHESLEEPYTBOASARVHWALKED  
ALONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTBESTHEOLDMVASVTO  
NTHEPARKWENCHFEXEDINGBREADCRGMBSTOTHEPIGEBASTHEAROMAOFYRESHLY  
YBAKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMIL  
YGAZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIG  
GLEDOITHDELIGHTASTHEYCHVSEXCOLORFULBGTTERYLIESINTHEGARDENTHEMA  
IESTICMOUNTAINSXSXODTALXLTHEIRPEAKSCOVER

-1407.2386195204908

[[ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'V', 'W', 'X', 'Y', 'Z'], [ 'H', 'T', 'K', 'L', 'M'], [ 'N', 'O', 'P', 'Q', 'U'] ]

THESUNROSESLBOLQCASTINGAWARMGLBOOVERTHESLEEPYTBOASARVHWALKED  
ALONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTBESTHEOLDMVASVTO  
NTHEPARKWENCHFEXEDINGBREADCRGMBSTOTHEPIGEBASTHEAROMAOFYRESHLY  
YBAKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMIL

YGAZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIG  
GLEDOITHDELIGHTASTHEYCHVSEXCOLORFULBGTTERYLIESINTHEGARDENTHEMA  
IESTICMOUNTAINSXSTOXODTALXLTHEIRPEAKSCOVER

-1386.9375895248259

-1386.9375895248259

[[ 'H', 'T', 'K', 'L', 'M'], [ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'N', 'O', 'P', 'Q', 'U'], [ 'V', 'W', 'X', 'Y',  
'Z']]

THESUNROSESLOIFLCASTINGAWARMGLOIOVERTHESLEEPYTOINSARVAWALKEDAL  
ONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTOESTHEOLDMVNSVTONT  
HEPARKWENCHFEXEDINGBREADCRUGBSTOTHEPIGEONSTHEAROMAOFYRESHLYB  
AKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMILYG  
AZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIGGL  
EDIBTHDELIGHTASTHEYCAVSEXCOLORFULBUTTERYLIESINTHEGARDENTHEMABE  
STICMOUNTAINSXSTOXODTALXLTHEIRPEAKSCOVER

-1386.9375895248259

[[ 'H', 'T', 'K', 'L', 'M'], [ 'S', 'E', 'C', 'R', 'T'], [ 'A', 'B', 'D', 'F', 'G'], [ 'N', 'O', 'P', 'Q', 'U'], [ 'V', 'W', 'X', 'Y',  
'Z']]

THESUNROSESLOIFLCASTINGAWARMGLOIOVERTHESLEEPYTOINSARVAWALKEDAL  
ONZTHEWEACHFEXELINZTHESOFTSVNDWETWEXENHERTOESTHEOLDMVNSVTONT  
HEPARKWENCHFEXEDINGBREADCRUGBSTOTHEPIGEONSTHEAROMAOFYRESHLYB  
AKEDBREADWAFTEDTHROUGHTHEAIRMAKINGEVERYONESMOUTHSWATEREMILYG  
AZEDATTHENIGHTSKYMESMERIZEDBYTHETWINKLINGSTARSTHECHILDRENGIGGL  
EDIBTHDELIGHTASTHEYCAVSEXCOLORFULBUTTERYLIESINTHEGARDENTHEMABE  
STICMOUNTAINSXSTOXODTALXLTHEIRPEAKSCOVER