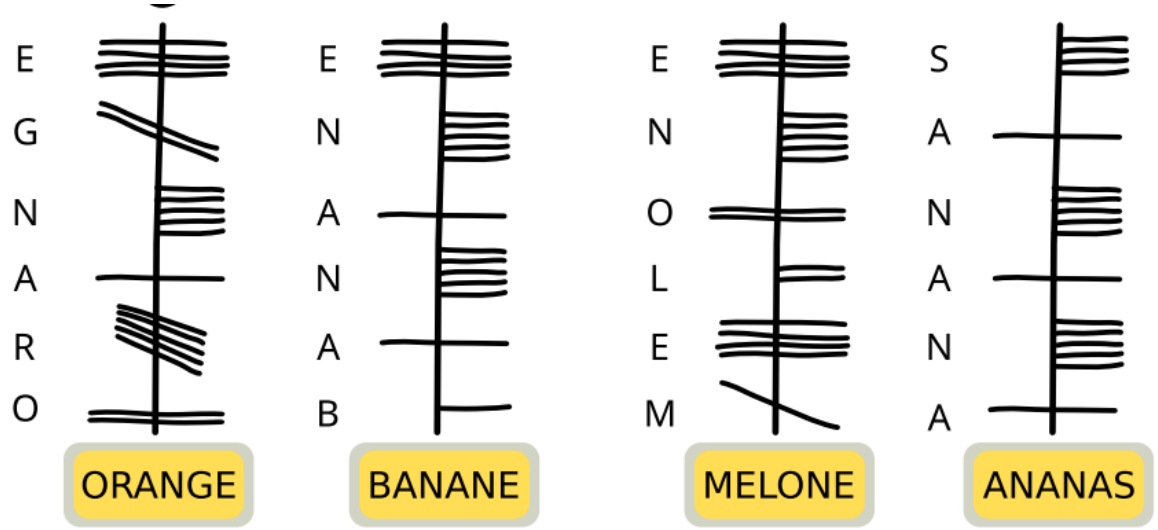
Vorbereitungsaufgabe 1: Versuchen Sie, die Wörter in lateinischer Schrift den Wörtern in der altirischen Ogham-Schrift zuzuordnen[1]. Überlegen Sie dabei, welche Eigenschaften Alphabete haben müssen, damit dies überhaupt gelingen kann.



Offensichtlich erfolgt die Anordnung der Buchstaben im Ogham-Alphabet vertikal an einem Strich. Das jeweils oberste Element ist 3 x gleich, 1 x verschieden, also entspricht es dem letzten Buchstaben der Wörter (-e, -e, -e, -s) und es wird von unten nach oben geschrieben. Weil Ananas also das 4. Wort sein muss, glückt die Entzifferung von a und n auf Anhieb. Durch Einsetzen der Symbole in den anderen Wörtern ergeben sich auch die anderen.

- *Buchstaben kodieren Laute linear in getrennter Abfolge*
- *Der Lautwert eines Buchstabens ist konstant und unabghängig von seiner Position*
- *Die Zuordnung zwischen Buchstaben und Laut ist eindeutig und unveränderlich*

In der Praxis gibt es Schriften, die zu allen 3 Erkenntnissen Ausnahmen haben. So können Buchstaben im Arabischen oder Indischen teils über- oder ineinander geschrieben werden, oder es werden Zeichen nicht für Buchstaben, sondern für ganze Silben verwendet.

(Optional: Vergleichen Sie dies mit ihrem Wissen über die mathematische Funktionenlehre: entspricht die Zuordnung von Buchstaben und Lauten einer injektiven oder gar bijektiven Funktion? Überlegen sie anhand des Deutschen „ü, y“ oder des Englischen „enough“).

[1] Biber-Wettbewerb 2023, SVIA

Vorbereitungsaufgabe 2: Es ist die streng fixierte Reihenfolge der Buchstabe in unserem Alphabet. Sie ist die Voraussetzung, dass Regeln für eine Transposition bei vertretbarer Gedächtnisleistung angewandt werden können. Zwar kann man das Alphabet in unterschiedlicher Richtung in die Tabelle füllen, wie wir in der Aufgabe 2.2, Seite 78 gesehen haben. Es genügen aber auch dann bereits wenige Hinweise, um die Art der Anordnung zu erraten.

Wir halten fest: Von den 26! Möglichkeiten, die 26 Buchstaben unseres Alphabets anzuordnen nutzen die historischen Kryptosysteme beim Chiffrieren stets die (allen bekannte) Anordnung, wie sie überliefert ist und auch im ASCII-Code fixiert ist. Auf dieser Matrix werden sämtliche Algorithmen zur Transposition angewandt. Das ist leider nicht nur für die Verschlüsselung praktisch, sondern auch für die Kryptoanalyse.

Linguistischer Exkurs zu Schriften: Sprachliche und kulturelle Informationen zu Schriften als Ergänzung zum geschichtlichen und gesellschaftlichen Kontext, Seite 52:

Mindestens in Ägypten, Mesopotamien, Südosteuropa (Vinca, Linear), China, Mittelamerika (Maya) und im Indus-Tal sind Schriften unabhängig von einander entwickelt worden, interessanterweise etwa zur gleichen Zeit, nämlich nach der Sesshaftwerdung (neolithische Revolution), als das Bedürfnis kam, komplexere Gesellschaften zu verwalten. Als Symbole dienten naturalistische Darstellungen von Tieren und Gegenständen (Ägypten, Sumer, Alteuropa) oder religiöse Symbole wie Götterdarstellungen (China, Maya). Die Symbole vereinfachten sich allmählich und wurden zu Lautabstraktionen. Sie wurden zur Schreibung von Wörtern benutzt, in denen gleich oder ähnlich lautende Silben vorkommen (Syllabar). Der jüngste Abstraktionsschritt ist dann die Akronymie, d.h. ein Zeichen bedeutet nur noch den Anfangsbuchstaben des Wortes, womit sich das Repertoire an Buchstaben auf das Lautinventar der Sprache (15-100, mit starker Tendenz zu einem Mittelwert um 20-40) reduzierte. Dies ermöglichte, dass weite Teile der Bevölkerung schriftkundig werden konnten, während dies in den frühen Stadien Eliten oder Priestern vorbehalten war. Das Festhalten des Chinesischen am ideographischen Typ hat Gründe: In China werden nicht weniger als 13 dermassen verschiedene Sprachen gesprochen, dass eine Verständigung nicht möglich ist, jedoch über dieselbe Schrift durchaus, zudem ist das Lesetempo nicht unerheblich höher als bei Alphabet-Schriften. Ausserdem sind die sinitischen Sprachen Tonsprachen und Rekordhalter der Homonymie, d.h. ein Wort wie „shi“ kann je nach Kontext und Tonfall über 10 komplett verschiedene Bedeutungen haben.

Viele Schriften sind sekundär abgeleitet von einer anderen und für eine bestimmte Sprache optimiert worden. Die christlichen Missionare haben so einige dieser Schriften für ihre Zielsprachen entwickelt: Das Kyrillische Alphabet (aus dem Griechischen), Inuktitut, Cree (frei erfunden). Alle südasiatischen Brahmi-Schriften von Sri Lanka bis Tibet leiten sich weit

entfernt aus der mesopotamischen Aramäisch-Variante ab. Wiederum andere sind originelle Neuschöpfungen nach bestehenden Vorbildern, etwa das Georgische oder das Koreanische.

Das indische Devanagari: Als Nachfahre der Brahmi-Schrift des 3. Jh. v.Chr. ist sie heute die vierthäufigste Schrift der Welt. Ursprünglich für Sanskrit, die Sprache der buddhistischen Texte, wird sie heute in Dutzenden Sprachen Südasiens verwendet. Es ist eine Art Segmentschrift, in der die Vokale diakritisch angedeutet werden, und das kurze a überhaupt nicht geschrieben wird. Dennoch existieren Buchstaben für alle Laute.

Das Inuktitut-Syllabar: Mitte des 19. Jh. haben mährische Missionare in Grönland und Labrador Evangelien zuerst in lateinischer Schrift, später in Cree (Algonquian-Amerindisch) gedruckt. Die eskimo-aleutischen Sprachen weichen aber dermassen ab von allen anderen, dass eine besser Lösung gefunden werden musste. Das Besondere an diesen Sprachen ist die Polysynthese: Ganze Satzaussagen werden zu einem einzigen Wort fusioniert mit unselbständigen Bausteinen. Eine Silbenschrift war nur schon für die bessere Lesbarkeit naheliegend. Sie wird heute nur in Kanada gebraucht.

Das japanische Katakana: Das Japanische hat 3 Schriften entwickelt und verwendet diese in allen Texten parallel. Wortstämme (Kanji) sind ideographisch und lehnen sich an die chinesischen Zeichen an. Endungen und grammatische Markierungen werden in der Silbenschrift Hiragana an das Ideogramm gehängt. Zum Teil werden seltene Kanji in Klammern in Hiragana geschrieben. Fremdsprachliche Termini oder ausländische Eigennamen stehen in der Silbenschrift Katakana. Die Zeichen beider Silbenschriften sind Vereinfachungen aus chinesischen Schriftzeichen. Beide Silbenschriften wären problemlos in der Lage, die ganze japanische Sprache zu verschriftlichen. Es gehört zu den erstaunlichsten Phänomenen, dass das Japanische bis heute an der Kombination der drei Systeme festhält..

Alle Alphabetschriften besitzen eine feste Reihenfolge ihrer Buchstaben, die (falls überliefert) als Merkhilfe und als Ordnungsprinzip diente. Viele machten sich diesen Umstand gar zunutze, um Zahlen mit Buchstaben zu schreiben, wie etwa im Altgriechischen (1 = Alpha, 2 = Beta etc). Die Logik dahinter ist sehr unterschiedlich. **In unseren phönizischen Alphabeten ist die Reihenfolge vollständig arbiträr,** und bei der Übernahme der Schrift von den Griechen durch die Etrusker, wurden Zeichen weggelassen, abgeändert und an beliebiger Stelle des Alphabets eingefügt, so z.B. das im Griechischen nicht vorhandene F.

Ganz anders sieht es im Devanagari aus. Die alten Inder, die fast alle Lebensbereiche zur Wissenschaft erhoben, ordneten die Laute systematisch nach sachlichen Kriterien: zunächst die Vokale, dann die Diphthonge, dann die Konsonanten ausgehend von ihrer Artikulationsstelle im Rachenraum von hinten (k) nach vorne (p).

Vergleich der Alphabete in Gruppen (Diskussion der Ergebnisse)

- In Silbenschriften erfolgt die Anordnung als Matrix mit den Vokalen, gefolgt von Reihen mit unterschiedlichen Stützkonsonanten
- Nicht alle Alphabete besitzen gleich viele Buchstaben
- Gewisse Laute, wie F oder die stimmhaften Konsonanten d, g, sind offenbar seltener als andere
- Es gibt immer deutlich mehr Konsonanten als Vokale
- Die Anordnung des lateinischen Alphabets wirkt im Vergleich besonders arbiträr

Latein	A	B	C	D	E	F	G	H	I	J	K	L	M
Indisch	A	I	U	R	L	E	O	M	H	K	G	N	C
Inuit	I	U	A	H	P	T	K	G	M	N	S	L	J
Katakana	A	I	U	E	O	K	S	T	N	H	M	Y	R

Latein	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Indisch	J	T	D	P	F	B	Y	V	S	Q	W	X	Z
Inuit	V	R	Q	C	B	D	E	F	O	W	X	Y	Z
Katakana	W	G	Z	D	B	P	C	F	J	L	Q	V	X

- Bei regelmässiger Alphabetabfolge würde ein Trigramm frühestens wieder nach kgV(Schlüssellänge, Anzahl Alphabete) gefunden, falls dort dann eines wäre. Es bräuchte wohl sehr lange Texte, um ein solches ausfindig zu machen.
- Sie sollten möglichst kein Vielfaches voneinander sein, aus oben genanntem Grund
- Natürlich vorkommende Folgen wie die Fibonacci-Folge, die Primzahlen oder auch die Kommastellen von π könnten so behandelt werden, dass jedes Element der Reihe modulo 4 gerechnet wird (da es ja 4 Alphabete sind). Wenn x %4 == 0‘: Latein, wenn x %4 == 1: Indisch, wenn x %4 == 2: Inuit, wenn x %4 == 3: Japanisch.
    - So kann die Sicherheit nochmals deutlich erhöht werden, da die Chiffrierungsgrundlage durch eine weitere Abstraktionsebene verschleiert wird.

Da die Fibonacci-Folge auch eine hervorragende Programmierübung darstellt, wollen wir diesen Ansatz versuchen.

Lösungsvorschlag zum Programm

```
import math

def n_te_fibonacci(n):
    f_n = (1/math.sqrt(5))*((math.pow((1+math.sqrt(5))/2,n)) - math.pow((1-math.sqrt(5))/2, n))
    return int(f_n)

def encrypt(Klartext, key):
    key_length = len(key)
    key_als_int = [ord(i) for i in key]
    alpha_latein = ['a', 'b', 'c', 'd','e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
'x', 'y', 'z']
    alpha_devanagari = ['a', 'i', 'u', 'r', 'l', 'e', 'o', 'm', 'h', 'k', 'g', 'n', 'c', 'j', 't', 'd', 'p', 'f', 'b', 'y', 'v',
's', 'q', 'w', 'x', 'z']
    alpha_inuit = ['i', 'u', 'a', 'h', 'p', 't', 'k', 'g', 'm', 'n', 's', 'l', 'j', 'v', 'r', 'q', 'c', 'b', 'd', 'e', 'f', 'o', 'w',
'x', 'y', 'z']
    alpha_katakana = ['a', 'i', 'u', 'e', 'o', 'k', 's', 't', 'n', 'h', 'm', 'y', 'r', 'w', 'g', 'z', 'd', 'b', 'p', 'c', 'f', 'j',
'l', 'q', 'v', 'x']

    Cyphertext = ""
    for i in range(len(Klartext)):   # man beachte, dass i mit 0 beginnt. Das ist aber kein
                                     # Problem. Oft wird die Folge
        if n_te_fibonacci(i) % 4 == 0:  # mit fakultativem zusätzlichen 0 am Anfang verwendet:
                                        # Moivre-Binet ergibt dann auch 0.
            Verschiebung = key_als_int[i % len(key)] - 65
            ciphre = (alpha_latein.index(Klartext[i]) + Verschiebung) % 26
            Cyphertext += alpha_latein[ciphre]
        if n_te_fibonacci(i) % 4 == 1:
            Verschiebung = key_als_int[i % len(key)] - 65
            ciphre = (alpha_devanagari.index(Klartext[i]) + Verschiebung) % 26
            Cyphertext += alpha_devanagari[ciphre]
        if n_te_fibonacci(i) % 4 == 2:
            Verschiebung = key_als_int[i % len(key)] - 65
            ciphre = (alpha_inuit.index(Klartext[i]) + Verschiebung) % 26
            Cyphertext += alpha_inuit[ciphre]
```

```
        if n_te_fibonacci(i) % 4 == 3:
            Verschiebung = key_als_int[i % len(key)] - 65
            ciphre = (alpha_katakana.index(Klartext[i]) + Verschiebung) % 26
            Cyphertext += alpha_katakana[ciphre]
    return Cyphertext

if __name__ == '__main__':
    Klartext = input("Klartext:").lower()
    key = input("Key:").lower()
    print(encrypt(Klartext, key))
```

Verschlüsselung der ersten beiden Abschnitte der Zusammenfassung Seite 77

Tgzgwykdmspeizlowdlpmbpquioxbkdkoieckdbqwdyebyydtdzowdiegdsbmnkmykiqcpyslwrlwar
cxtsiabcfdsbykyohyopurvunihqixomccgxbmvdidswjwmdniididcnqrvukpjcmuliidixdekchuxxusp
uxohitjykhktxsitsimswiorimlqpxtowuxxmevvczexohitjywocxuwidertuvozxekrqvciohyoiddaymob
erwfsdwijrstorpeqrbisridfsduvozxeccidicortsiqopjowjorwolusqimlhsjjortovqxxyuircmubxuxekp
wklwjsxkdmexidertdvqxwfywydmexidnmunejorisgxovxomjleisihditkqqvwqejtsiwolusqxkpjerwn
ihqedjidqixomccgxbmvdauclqvftsiwolusqimlhsjjxmsrximlhsjjvmsrekpfugexbxmovtortevvdikxhi
yqydekcauxhyqkuvihxxmovtorcewidi

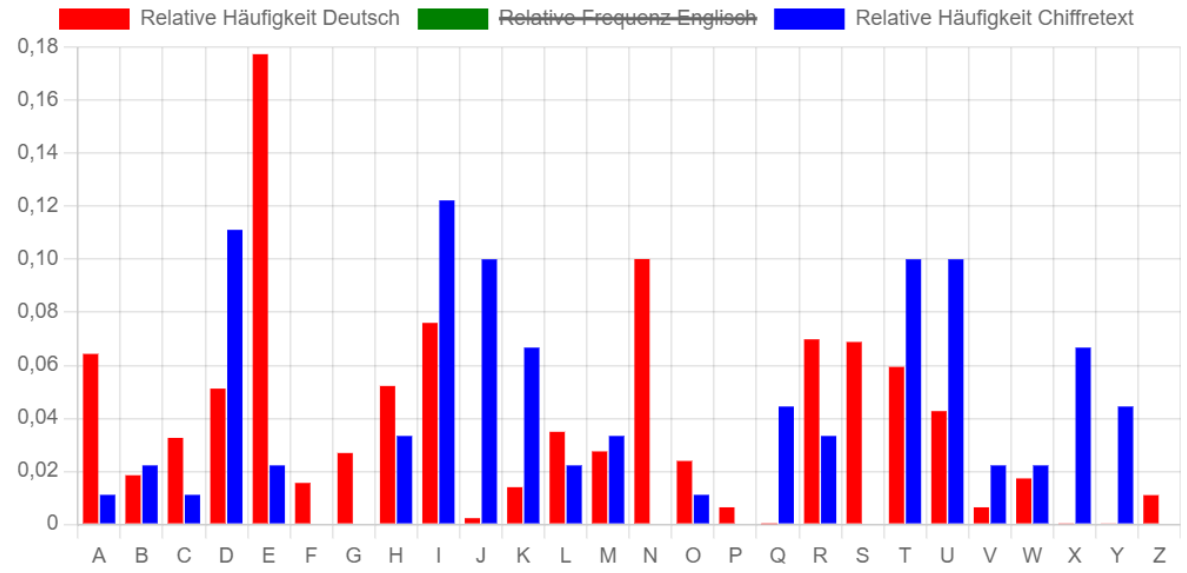
Programm zur Friedman’schen Charakteristik:

```
from collections import Counter
def friedman (text):
    freq = countocc(text)
    sum = 0
    for letter_freq in freq:
        sum += (letter_freq - (1/26))**2
    return sum

def countocc(eingabe):
    length = len(eingabe)
    counter = Counter(eingabe)
    alphabet = []
    for i in range(26):
```

```
        letter = chr(97+i)
        occ = counter[letter]
        alphabet.append(occ/length)
    return alphabet

if __name__ == '__main__':
    eingabe = input('Text:'). lower()
    print(friedman(eingabe))
```

Ergebnis für die Friedman'sche Charakteristik: 0.0102

Visualisierung der Buchstabenhäufigkeit mit
https://cryptbreaker.marcwidmer.xyz/solve



Fazit:

- Schriften sind arbiträr aufgebaut in Struktur und Buchstabenreihenfolge. Sie sind wesentlich von gesellschaftlicher Konvention ihrer Herkunftskultur geprägt. Alphabete sind konservativ und gegenüber Veränderungen über Jahrtausende stabil. Wir orientieren uns an ihnen beim Denken und Strukturieren, oft ohne uns dessen bewusst zu sein.
- Es ist ausreichend zu wissen, welche Alphabete zugrunde gelegt werden, um die alternative Reihenfolge zu rekonstruieren. Dieses Wissen wird somit zum Teil des Schlüssels. Er muss nicht zwingend auswendig gelernt werden, und wenn, dann hat man Teile eines weiteren nützlichen Alphabets gelernt.
- Bei geschickter Wahl der Schlüssellänge kann das Verfahren bereits bei einem so kurzen Schlüssel wie „key“ die stochastischen Auffälligkeiten im verschlüsselten Text in hohem Masse auflösen und verschleiern. Es bedingt allerdings, dass neben dem Schlüssel auch die verwendeten Alphabete und ihre Reihenfolge bekannt sind.
- Mit der Wahl der Fibonacci-Folge kann zudem die Reihenfolge, in der die relevanten Alphabete zum Einsatz kommen, zusätzlich verschleiert werden.