Diana Cryptosystem

During the Vietnam War "clandestine" communication between US military forces was based on the Diana Cryptosystem. This methode to encode en decode messages is theoretically unbreakable when used properly. The cryptographic cipher is based on two techniques.

First of all, it uses a **trigraph** to convert two letters into a third letter. This conversion uses a fixed lookup table — an example of which is pictured below — that maps each combination of two black letters into a third red letter. What is specific about this conversion is that for each triplet of letters it holds that any two letters of the triplet are always converted into the third letter of the triplet. For example, if it is given that the letters B and K are converted in the letter O, then we also know that the letters K and B yield the letter O, and also that the letters O and B yield the letter K, and that the letters K and B yield the letter O. Actually, it is quite easy to compute the conversion. If we find the two given letters in the alphabet at positions p_1 and p_2 (where A is at position 0, B is at position 1, and so on), then we find the third letter in the alphabet at position $(25 - p_1 - p_2)\ \textrm{mod}\ 26$, where the operator $\textrm{mod}$ represents the remainder after integer division.

In addition, the Diana Cryptosystem makes use of a so-called **one-time pad**. This is nothing more than a randomly generated list of letters. To improve readability the letters are usually displayed in groups of five letters, but in general each character in the one-time pad that is not a letter may be ignored (also including the spaces used for formatting the groups). As an example

we consider the following one-time pad.

WHTVI AUCFU RETFK OMSAL
MYMNE ZIEGP UKVTF WZHOK
GORWY WETFR COYET OOWHY
ZPDDA CMMXT VYTJI RRQGU
VAXPM IPIXU QUXIP MAXIU

To encode the message ATTACK AT DAWN, we proceed as follows. First, a random n-letter fragment is selected from the one-time pad, where $n \in \mathbb{N}_0$ is fixed as part of the cryptosystem. Say, for example, that we have chosen the ten letters UKVTF WZHOK. The letters of the message (all characters that are not letters must be ignored) are then written under the letters of the one-time pad that follow the randomly selected fragment. Thereafter each pair of letters (a letters from the one-time pad and a letter from the original message at the same position) are converted into a third letters using the trigraph lookup table. The resulting letters form the encrypted message.

one-time pad: UKVTF WZHOK **GORWY WETFR CO**YET OOWHY
origineel bericht: **ATTAC KATDA WN**
---
gecodeerd bericht: **TSPDZ TVNRI BY**

The text UKVTF WZHOK **TSPDZ TVNRI BY** is transmitted by morse code: the letters of the randomly selected fragment are transmitted unencrypted and the letters of the message are transmitted encrypted. Because of the symmetric nature of the trigraph, decoding a message follows the exact same procedure as encoding a message. First the recipient looks up the first ten letters of the message in the one-time pad (the recipient must use the same one-time pad as the sender), and decodes the remaining letters by again making use the trigraph lookup table.

one-time pad: UKVTF WZHOK **GORWY WETFR CO**YET OOWHY
gecodeerd bericht: **TSPDZ TVNRI BY**
---
origineel bericht: **ATTAC KATDA WN**

A military of the US forces that was fighting in the Vietnam War explains one use case of the Diana Cryptosystem in the following way:

> *Special Forces were one of (if not the only) units in Vietnam to utilize Morse code on a regular basis. We used a method of encryption called the Diana Cryptosystem.*
>
> *The basis of these* one-time pads*, is that there were only two matching pads in existence, and they would only be used one time. They were booklets that contained randomly generated groups of 5-letter* words*, 30 words to a page. The person sending a message would first write the letters to the message, over these random groups of words. Included in the front of each one-time pad was a one-page encryption table. If I wanted to send the letter P, and the letter under the P was an A, then I would send a K. The person listening on the frequency at the other end, would have the other matching pad. They would write the letter they received (a K) over the letter in their one-time pad (an A), and decipher it based on the table, yielding the original letter P.*
>
> *Each communication site in Vietnam (we had over 100 A-Camps along the*

Cambodian / Laotian border, and some 20 B-detachment sites spread over the country) had a different pad, depending on the location they were having the commo-check with. It obviously was very important that both people were using the appropriate matching pads, or the deciphered messages would not make any sense.

After a while, most of us became so proficient with the system, that we actually learned the deciphering matrix by heart. No matter what pads anyone had, the combinations always were the same. i.e. Any 3 letters always went together, regardless of the order; BKO/KOB/OBK/BOK. After listening to thousands and thousands of transmissions, it really got quite simple. If I was listening to code, and a letter B was sent (now remember, we usually sent around 20-25 words *(5 letters per word) a minute, hence the importance of the* speed *keys!), and the letter it was associated with was an O, most of us would decipher as we heard it, and just write the K. That may sound like quite a yarn, but it is absolutely true.*

The combination of a trigraph lookup table and a one-time pad may seem like a neat parlor trick, but the resulting cryptographic cipher is actually quite strong. In fact, assuming that the pads are truly randomly generated, never reused and never compromised the system is unbreakable. It therefore comes as no surprise that was and still is used by many intelligence agencies around the world. The KGB often issued its agents one-time pads printed on tiny sheets of *flash paper* — paper chemically converted to nitrocellulose, which burns almost instantly and leaves no ash.

Assignment

Define a class Diana that can be used to encode and decode messages according to the Diana Cryptosystem using a given one-time pad. This class must support at least the following methods:

- An initialisation methat __init__ that takes the location of a text file. The text file must contain the letters of a one-time pad. The text file may contain multiple lines, and apart from letters it may also contain other characters. The one-time pad itself is constructed from the letters in the file, ignoring all other characters.
- A method index that takes a string. The method must reduce the given string into a sequence of letters (ignoring all characters that are not letters), and must lookup the first occurrence of this sequence of letters in the one-time pad. In looking up the fragment of the one-time pad, not distinction should be made between uppercase and lowercase letters. If the sequence of letters occurs in the one-time pad, the method must return the position of the first letter that follows the sequence of letters in the one-time pad. We assume that the first letter of the one-time pad is at position 0, the second letter at position 1, and so on. In case the sequence of letters does not occur in the one-time pad, the method must raise an AssertionError with the message invalid prefix.
- A method trigraph that takes two strings that each consist of a single letter. The method must return the uppercase letter that is found using the trigraph lookup table with the two given letters.
- A methode encode that takes a string. The given string must contain a sequence of letters that occur in the one-time pad, followed by the letters of a message that must be encoded according to the Diana Cryptosystem. The message may contain both uppercase and lowercase letters. Apart from letters, the string may also contain other characters that should be ignored by the encoding procedure. The message also has an optional second

parameter that takes an integer n that indicates how many letters of the given string must be used as the fragment that is looked up in the one-time pad (default value: $n = 10$). The method must return the encoded message, that only contains uppercase letters. In case the first n letters of the given string are not found in the one-time pad, the method must raise an AssertionError with the message invalid prefix. In case the first occurrence of the first n letters of the given string in the one-time pad is followed by fewer letters than there are letters in the given message, the method must raise an AssertionError with the message one-time pad is too short.

- A method decode that works in exactly the same way as the method encode, so that is can be used to decode encrypted messages. We state again that the Diana Cryptosystem is conceived such that encoding and decoding work according to the same procedure.

Example

In the following interactive session we assume that the text file otp.txt is located in the current directory.

```
>>> diana = Diana('otp.txt')

>>> diana.index('UKVTF WZHOK')
40
>>> diana.index('CMMXT VYTJI RRQGU')
80
>>> diana.index('ABCDE FGHIJ')
Traceback (most recent call last):
AssertionError: invalid prefix

>>> diana.trigraph('Q', 'K')
'Z'
>>> diana.trigraph('t', 'f')
'B'

>>> diana.encode('UKVTF WZHOK attack at dawn')
'TSPDZTVNRIBY'
>>> diana.encode('CMMXT VYTJI RRQGU meet at ten tonight', 15)
'SVYRNYRNPMVSULDU'
>>> diana.encode('ABCDE FGHIJ zero dark thirty')
Traceback (most recent call last):
AssertionError: invalid prefix
>>> diana.encode('VAXPM IPIXU zero dark thirty')
Traceback (most recent call last):
AssertionError: one-time pad is too short

>>> diana.decode('UKVTF WZHOK TSPDZ TVNRI BY')
'ATTACKATDAWN'
>>> diana.decode('CMMXT VYTJI RRQGU SVYRN YRNPM VSULD U', 15)
'MEETATTENTONIGHT'
>>> diana.decode('ABCDE FGHIJ zero dark thirty')
Traceback (most recent call last):
AssertionError: invalid prefix
>>> diana.decode('VAXPM IPIXU zero dark thirty')
Traceback (most recent call last):
AssertionError: one-time pad is too short
```

Tijdens de Vietnamoorlog maakte het Amerikaanse leger gebruik van het **Diana Cryptosystem**. Deze methode om berichten te coderen en decoderen kan in theorie niet gekraakt worden indien

ze volgens de regels van de kunst gebruikt wordt. De versleuteling is gebaseerd op twee technieken.

Eerst en vooral wordt er gebruik gemaakt van een **trigraph** om twee letters om te zetten naar een derde letter. Deze omzetting gebeurt aan de hand van een vaste tabel — waarvan hieronder een voorbeeld gegeven wordt — waarbij twee zwarte letters telkens een rode letter opleveren. Typisch aan deze omzetting is dat voor elk drietal letters geldt dat twee van de drie letters door de trigraph steeds worden omgezet in de derde letter. Als we bijvoorbeeld weten dat de letters B en K de letter O opleveren, dan leveren ook K en B de letter O op, maar leveren ook K en O de letter B op, en leveren ook O en B de letter K op. Eigenlijk kan de omzetting ook makkelijk berekend worden. Als we de twee gegeven letters in het alfabet terugvinden op posities p_1 en p_2 (waarbij A op positie 0 staat, B op positie 1, enzoverder), dan vinden we de derde letter in het alfabet terug op positie $(25 - p_1 - p_2)\ \textrm{mod}\ 26$. Hierbij staat de operator $\textrm{mod}$ voor de rest na gehele deling.

Daarnaast wordt gebruik gemaakt van een **one-time pad**. Dit is eigenlijk niets anders dan een willekeurige opeenvolging van letters. Voor de leesbaarheid worden deze vaak weergegeven in groepen van vijf letters, maar elk karakter dat geen letter is moet genegeerd worden in het one-time pad (dus ook de spaties). Als voorbeeld beschouwen we het volgende one-time pad.

WHTVI AUCFU RETFK OMSAL
MYMNE ZIEGP UKVTF WZHOK
GORWY WETFR COYET OOWHY
ZPDDA CMMXT VYTJI RRQGU

Om dan bijvoorbeeld het bericht ATTACK AT DAWN te coderen, gaan we als volgt te werk. Eerst wordt er een willekeurig fragment van een vooraf afgesproken lengte gekozen uit het one-time pad. Stel bijvoorbeeld dat we kiezen voor de tien letters UKVTF WZHOK. De letters van het bericht (alle karakters die geen letter zijn worden genegeerd) worden dan onder de letters van het one-time pad geschreven die volgen op het willekeurig gekozen fragment. Daarna wordt elk paar letters (een letter uit het one-time pad en een letter uit het origineel bericht op dezelfde positie) omgezet naar een derde letter door gebruik te maken van een trigraph. Deze laatste letters leveren uiteindelijk het gecodeerde bericht op.

one-time pad: UKVTF WZHOK **GORWY WETFR CO**YET OOWHY
origineel bericht: **ATTAC KATDA WN**
---
gecodeerd bericht: **TSPDZ TVNRI BY**

Het bericht UKVTF WZHOK **TSPDZ TVNRI BY** wordt dan verstuurd in morsecode. Hierbij wordt dus eerst het willekeurige fragment van tien letters in niet-gecodeerde vorm verstuurd, gevolgd door het gecodeerde bericht zelf. Omwille van de symmetrie van de trigraph werkt het decoderen van een bericht volgens exact dezelfde procedure als het coderen. Eerst zoekt de ontvanger de eerste tien letters van het bericht op in het one-time pad (de ontvanger moet hiervoor hetzelfde one-time pad gebruiken als de verzender), en decodeert het dan opnieuw door gebruik te maken van een trigraph.

one-time pad: UKVTF WZHOK **GORWY WETFR CO**YET OOWHY
gecodeerd bericht: **TSPDZ TVNRI BY**
---
origineel bericht: **ATTAC KATDA WN**

Een soldaat die meestreed tijdens de Vietnamoorlog verwoordde het gebruik van het Diana Cryptosystem op de volgende manier:

> *Special Forces were one of (if not the only) units in Vietnam to utilize Morse code on a regular basis. We used a method of encryption called the Diana Cryptosystem.*
>
> *The basis of these* one-time pads*, is that there were only two matching pads in existence, and they would only be used one time. They were booklets that contained randomly generated groups of 5-letter* words*, 30 words to a page. The person sending a message would first write the letters to the message, over these random groups of words. Included in the front of each one-time pad was a one-page encryption table. If I wanted to send the letter P, and the letter under the P was an A, then I would send a K. The person listening on the frequency at the other end, would have the other matching pad. They would write the letter they received (a K) over the letter in their one-time pad (an A), and decipher it based on the table, yielding the original letter P.*
>
> *Each communication site in Vietnam (we had over 100 A-Camps along the Cambodian / Laotian border, and some 20 B-detachment sites spread over the country) had a different pad, depending on the location they were having the commo-check with. It obviously was very important that both people were using the appropriate matching pads, or the deciphered messages would not make any sense.*

> *After a while, most of us became so proficient with the system, that we actually learned the deciphering matrix by heart. No matter what pads anyone had, the combinations always were the same. i.e. Any 3 letters always went together, regardless of the order;* BKO/KOB/OBK/BOK*. After listening to thousands and thousands of transmissions, it really got quite simple. If I was listening to code, and a letter* B *was sent (now remember, we usually sent around 20-25* words *(5 letters per word) a minute, hence the importance of the* speed *keys!), and the letter it was associated with was an* O*, most of us would decipher as we heard it, and just write the* K*. That may sound like quite a yarn, but it is absolutely true.*

De combinatie trigraph en one-time pad levert een zeer sterke vorm van encryptie op. In de veronderstelling dat de letters van het pad echt willekeurig gegenereerd worden, nooit hergebruikt worden en niet gecompromiteerd zijn, kan aangetoond worden dat de code onbreekbaar is. Het is daarom niet verwonderlijk dat heel wat inlichtingendiensten gebruik maakten en maken van deze vorm van encryptie. De KGB gaf aan haar agenten bijvoorbeeld vaak one-time pads mee die gedrukt waren op *flash paper* — papier dat chemisch omgezet werd naar nitrocellulose, waardoor het bijna onmiddellijk verbrandt zonder as na te laten.

Opgave

Definieer een klasse Diana waarmee berichten kunnen gecodeerd en gedecodeerd worden volgens het Diana Cryptosystem met een vooraf vastgelegd one-time pad. Deze klasse moet minstens de volgende methoden ondersteunen:

- Een initialisatiemethode __init__ waaraan de locatie van een tekstbestand moet doorgegeven worden. Dit tekstbestand moet de letters van een one-time pad bevatten. Hierbij mag het tekstbestand bestaan uit meerdere regels, en naast letters ook andere karakters bevatten. Het one-time pad zelf wordt opgebouwd uit de opeenvolgende letters in het bestand, waarbij de andere karakters genegeerd worden.
- Een methode index waaraan een string moet doorgegeven worden. De methode moet de gegeven string herleiden tot een reeks opeenvolgende letters (alle karakters die geen letter zijn moeten hierbij dus genegeerd worden), en moet het eerste voorkomen van deze reeks letters opzoeken in het one-time pad. Bij het opzoeken mag geen onderscheid gemaakt worden tussen hoofdletters en kleine letters. Indien de reeks letters voorkomt in het one-time pad, dan moet de methode de positie teruggeven van de eerste letter die volgt op de reeks letters in het one-time pad. Hierbij staat de eerste letter van het one-time pad op positie 0, de tweede letter op positie 1, enzoverder. Indien de reeks letters niet voorkomt in het one-time pad, dan moet de methode een AssertionError opwerpen met de boodschap ongeldige prefix.
- Een methode trigraph waaraan twee strings moeten doorgegeven worden die elk bestaan uit één enkele letter. De methode moet de hoofdletter teruggeven waarop de twee gegeven letters worden afgebeeld volgens een trigraph.
- Een methode codeer waaraan een string moet doorgegeven worden. De gegeven string moet bestaan uit een reeks letters die voorkomen in het one-time pad, gevolgd door de letters van een bericht dat moet gecodeerd worden volgens het Diana Cryptosystem. Het bericht mag zowel hoofdletters als kleine letters bevatten. Naast letters mag de gegeven string ook andere karakters bevatten, die echter moeten genegeerd worden bij het coderen. De methode heeft ook nog een tweede optionele parameter waaraan een integer n kan

doorgegeven worden die aangeeft hoeveel letters van de string moeten gebruikt worden voor het opzoeken in het one-time pad (standaardwaarde: $n=10$). De methode moet het gecodeerde bericht teruggeven, dat enkel mag bestaan uit hoofdletters. Indien de eerste n letters van de gegeven string niet teruggevonden worden in het one-time pad, dan moet de methode een AssertionError opwerpen met de boodschap ongeldige prefix. Indien er na het voorkomen van de eerste n letters van de gegeven string in het one-time pad minder letters volgen dan er letters zijn in het gegeven bericht, dan moet de methode een AssertionError opwerpen met de boodschap one-time pad is te kort.

- Een methode decodeer die op exact dezelfde manier werkt als de methode codeer, en dus kan gebruikt worden om gecodeerde berichten te decoderen. We herhalen hier nogmaals dat bij het Diana Cryptosystem het coderen en decoderen werkt volgens exact dezelfde procedure.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat het tekstbestand otp.txt zich in de huidige directory bevindt.

```
>>> diana = Diana('otp.txt')

>>> diana.index('UKVTF WZHOK')
40
>>> diana.index('CMMXT VYTJI RRQGU')
80
>>> diana.index('ABCDE FGHIJ')
Traceback (most recent call last):
AssertionError: ongeldige prefix

>>> diana.trigraph('Q', 'K')
'Z'
>>> diana.trigraph('t', 'f')
'B'

>>> diana.codeer('UKVTF WZHOK attack at dawn')
'TSPDZTVNRIBY'
>>> diana.codeer('CMMXT VYTJI RRQGU meet at ten tonight', 15)
'SVYRNYRNPMVSULDU'
>>> diana.codeer('ABCDE FGHIJ zero dark thirty')
Traceback (most recent call last):
AssertionError: ongeldige prefix
>>> diana.codeer('VAXPM IPIXU zero dark thirty')
Traceback (most recent call last):
AssertionError: one-time pad is te kort

>>> diana.decodeer('UKVTF WZHOK TSPDZ TVNRI BY')
'ATTACKATDAWN'
>>> diana.decodeer('CMMXT VYTJI RRQGU SVYRN YRNPM VSULD U', 15)
'MEETATTENTONIGHT'
>>> diana.decodeer('ABCDE FGHIJ zero dark thirty')
Traceback (most recent call last):
AssertionError: ongeldige prefix
>>> diana.decodeer('VAXPM IPIXU zero dark thirty')
Traceback (most recent call last):
AssertionError: one-time pad is te kort
```