Geheimschreiber

In summer 1940, Germany demanded access to Swedish telephone cables to send encoded messages from occupied Norway back to the homeland. Sweden acceded but tapped the lines and discovered that a new cryptographic system was being used. The T52 *Geheimschreiber* (or *G-schreiber* in short), with more than 800 quadrillion settings, was conveying top-secret information but seemed immune to a successful codebreaking attack.

The Siemens & Halske T52 — also known as the *Geheimfernschreiber* ("secret teleprinter") or *Schlüsselfernschreibmaschine* (SFM) — was a World War II German cipher machine and teleprinter produced by the electrical engineering firm Siemens & Halske. The instrument and its traffic were codenamed *Sturgeon* by British cryptanalysts.

The Swedish intelligence service assigned mathematician Arne Beurling to the task, giving him only a pile of coded messages and no knowledge of the mechanism that had been used to encode them. But after two weeks alone with a pencil and paper he announced that he had decoded the messages. He also described how a complementary machine could be built to decode the messages automatically. Thanks to his work, Swedish officials learned in advance of the impending invasion of the Soviet Union. Unfortunately, Stalin's staff disregarded their warnings. In his book The Codebreakers Bengt Beckman accounts of the exploit. In his foreword to the book, Peter Jones writes:

> *To this day no one knows exactly how Beurling reasoned during the two weeks he spent on the G-Schreiber. In 1976 he was interviewed about his work by a group from the Swedish military, and became extremely irritated when pressed for an explanation. He finally responded, "A magician does not reveal his tricks." It seems the only clue Beurling ever offered was the remark, cryptic itself, that threes and fives were important.*

Assignment

In the end, the procedure used by the T52 *Geheimschreiber* did not turn out to be al too complicated. It uses an alphabet of m symbols that are all different and have a fixed order. The positions of the symbols in the alphabet are indexed $0, 1, \ldots, m - 1$.

In addition, two integers a and b are used as the key for encoding and decoding. Each symbol at position x in the alphabet is encoded as the symbol at position $E(x)$ in the

alphabet, where $$E(x) = (ax + b)\ \textrm{mod}\ m \qquad (0 \leq x < m)$$ and the operator $\textrm{mod}$ results in the remainder after integer division. Each symbol at position x in the alphabet is decoded as the symbol at position $D(x)$ in the alphabet, where $$D(x) = a'(x - b)\ \textrm{mod}\ m \qquad (0 \leq x < m)$$ and a' is the multiplicative inverse of a modulo m. In other words, a' is the integer in the interval $[0, m[$ that satisfies the equation $$1 = aa'\ \textrm{mod}\ m$$ The multiplicative inverse of a only exists if a and m are coprime. This is the case if the greatest common divisor of a and m equals 1. If this is the case, there is only one integer in the interval $[0, m[$ that satisfies the equation for the multiplicative inverse.

When the Germans noticed that messages sent using the T52 could be easily decoded, they tried to make the encryption procedure more complex. This was done by not encoding a message once but twice. First a message was encoded by a T52 that used keys a_1 and b_1, and the result was encoded again by a T52 that used keys a_2 and b_2. This is only possible if the two machines use the same alphabet. From the observation that $$E_2(E_1(x)) = (a_2((a_1x + b_1)\ \textrm{mod}\ m) + b_2)\ \textrm{mod}\ m = (a_1a_2x + (a_2b_1 + b_2))\ \textrm{mod}\ m$$ it follows that this has the same effect as encoding the message only once by a T52 that used keys a_1a_2$ and $(a_2b_1 + b_2)$.

Your task is to define a class `T52` whose objects represent T52 *Geheimschreiber* machines that can be used to encode and decode messages according to the procedures outlined above. This class must support at least the following methods:

- An initialisation method `__init__` that takes three arguments: two integers a and b and an m-character string that contains the symbols of the alphabet used for encoding and decoding. If not all characters of the given alphabet are different, the method must raise an `AssertionError` with the message `alphabet has repeated symbols`. If a and m are not coprime, the method must raise an `AssertionError` with the message *a* and *m* are not coprime. Of course, the cursive fragments in this message need to be filled up with the values a and m. The `fractions` module of the Python Standard Library implements a function `gcd` that can be used to compute the greatest common divisor of two integers.
- A method `encodeSymbol` that takes a one-character string. If the given character occurs in the alphabet of symbols, the method must return the encoded symbol. Otherwise the method must return the given character itself.
- A method `decodeSymbol` that takes a one-character string. If the given character occurs in the alphabet of symbols, the method must return the decoded symbol. Otherwise the method must return the given character itself.
- A method `encode` that takes a string. The method must return the given string where all characters that occur in the alphabet of symbols are replaced by their encoded symbols. All characters that do not occur in the alphabet of symbols must be retained.
- A method `decode` that takes a string. The method must return the given string where all characters that occur in the alphabet of symbols are replaced by their decoded symbols. All characters that do not occur in the alphabet of symbols must be retained.

In should also be possible to add two objects of the class `T52` using the `+` operator. This should result in a new object of the class `T52` that represents a T52 *Geheimschreiber* machine that encodes messages in the same way as if we would first encode the message using the machine to the left of the `+` operator and then encode the result once more using the machine to the right of the `+` operator. If the two `T52` objects that are added do not use the same alphabet of symbols, an `AssertionError` must be raised with the message `alphabets are different`.

Example

```
>>> machine1 = T52(3, 5, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')

>>> machine1.encodeSymbol('G')
'X'
>>> machine1.encodeSymbol('S')
'H'
>>> machine1.encodeSymbol('-')
'-'

>>> machine1.decodeSymbol('X')
'G'
>>> machine1.decodeSymbol('H')
'S'
>>> machine1.decodeSymbol('-')
'-'

>>> machine1.encode('G-SCHREIBER')
'X-HLAERDIRE'
>>> machine1.decode('X-HLAERDIRE')
'G-SCHREIBER'

>>> machine2 = T52(17, 11, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
>>> machine2.encode('X-HLAERDIRE')
'M-AQLBOKROB'

>>> machine12 = machine1 + machine2
>>> machine12.encode('G-SCHREIBER')
'M-AQLBOKROB'

>>> T52(4, 5, 'ABCDEFGHIJKLMMLKJIHGFEDCBA')
Traceback (most recent call last):
AssertionError: alphabet has repeated symbols

>>> T52(4, 5, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
Traceback (most recent call last):
AssertionError: 4 and 26 are not coprime

>>> machine1 + T52(17, 11, 'abcdefghijklmnopqrstuvwxyz')
Traceback (most recent call last):
AssertionError: alphabets are different
```

Tijdens de zomer van 1940 eiste Duitsland toegang tot de Zweedse telefoonkabels om gecodeerde berichten te kunnen versturen vanuit het bezette Noorwegen naar het thuisland. Zweden kon niet anders dan toegeven, maar begon meteen de lijnen af te luisteren en ontdekte al snel dat er een nieuw cryptografisch systeem gebruikt werd. De T52 *Geheimschreiber* (of kortweg *G-schreiber*) — een machine met meer dan 8 biljard verschillende instellingen — verzond geheime informatie die maar niet gekraakt leek te kunnen worden.

De Siemens & Halske T52 — ook gekend als de *Geheimfernschreiber* of *Schlüsselfernschreibmaschine* (SFM) — was een codeermachine die Duitsland gebruikte tijdens de Tweede Wereldoorlog. De machine en zijn gecodeerde berichten kregen van de Britse geheime dienst de codenaam *Sturgeon*.

De Zweedse inlichtingendiensten gaven wiskundige Arne Beurling opdracht om de code te kraken op basis van slechts een handvol gecodeerde berichten en zonder enige voorkennis over het mechanisme dat gebruikt was bij het coderen. Na twee weken puzzelen met enkel potlood en papier slaagde Beurling erin de berichten te ontcijferen. Hij kon ook beschrijven hoe ze een complementaire machine konden bouwen om berichten te decoderen. Dankzij deze doorbraak konden Zweedse ambtenaren berichten onderscheppen waarin sprake was van een nakende invasie van de Sovjet-Unie. Helaas sloeg het personeel van Stalin hun waarschuwingen in de wind. Het boek The Codebreakers van Bengt Beckman brengt relaas van het hele verhaal. In zijn voorwoord voor het boek schrijft Peter Jones:

> *Tot op heden weet niemand welke redenering Beurling maakte tijdens die twee weken dat hij zijn zinnen zette op de G-schreiber. In 1976 werd hij door een afdeling van het Zweedse leger ondervraagd, en raakte daarbij zeer geïrriteerd toen ze hem onder druk zetten om zijn werkmethode uit de doeken te doen. Uiteindelijk antwoordde hij: "Een goochelaar legt toch ook niet uit hoe zijn trucs werken." Het lijkt erop dat de enige aanwijzing die Beurling ooit gaf, verscholen zat in de cryptische (of wat had je gedacht) opmerking dat drieën en vijven belangrijk waren.*

Opgave

Uiteindelijk bleek de procedure die de T52 *Geheimschreiber* gebruikte helemaal niet zo ingewikkeld. Er wordt gewerkt met een alfabet van m symbolen die allemaal verschillend zijn en een vaste volgorde hebben. De posities van de symbolen in het alfabet worden genummerd van 0 tot en met $m - 1$.

De sleutel die gebruikt wordt voor het coderen en decoderen bestaat verder uit twee natuurlijke getallen a en b. Elk symbool op positie x in het alfabet wordt gecodeerd als het symbool op positie $C(x)$ in het alfabet, waarbij $$C(x) = (ax + b)\ \textrm{mod}\ m \qquad (0 \leq x < m)$$ en de operator $\textrm{mod}$ geeft de rest na gehele deling. Elk symbool op positie x in het alfabet wordt gedecodeerd als het symbool op positie $D(x)$ in het alfabet, waarbij $$D(x) = a'(x - b)\ \textrm{mod}\ m \qquad (0 \leq x < m)$$ en a' staat voor de multiplicatieve inverse van a modulo m. Met andere woorden, a' is het natuurlijk getal uit het interval $[0, m[$ waarvoor geldt dat $$1 = aa'\ \textrm{mod}\ m$$ De multiplicatieve inverse van a bestaat enkel als a

en m copriem zijn. Dat is het geval als de grootste gemene deler van a en m gelijk is aan 1. In dat geval bestaat er ook maar één getal in het interval $[0, m[$ dat voldoet aan de voorwaarde van de multiplicatieve inverse.

Toen de Duitsers hoogte begonnen te krijgen van het feit dat de gecodeerde berichten van de T52 makkelijk konden gekraakt worden, probeerden ze die nog complexer te maken. Ze deden dat door een bericht niet één maar twee keer te coderen. Een eerste keer door een T52 met sleutels a_1 en b_1 en een tweede keer door een T52 met sleutels a_2 en b_2. Dit kan enkel als de twee machines hetzelfde alfabet gebruiken. Uit de vaststelling dat $$C_2(C_1(x)) = (a_2((a_1x + b_1)\ \textrm{mod}\ m) + b_2)\ \textrm{mod}\ m = (a_1a_2x + (a_2b_1 + b_2))\ \textrm{mod}\ m$$ volgt echter dat dit dezelfde codering is als deze die gemaakt wordt door één enkele T52 met sleutels a_1a_2 en $(a_2b_1 + b_2)$.

Gevraagd wordt om een klasse `T52` te schrijven waarvan de objecten T52 *Geheimschreiber* machines voorstellen waarmee berichten kunnen gecodeerd en gedecodeerd worden volgens de procedures die hierboven beschreven werden. Deze klasse moet minstens de volgende methoden ondersteunen:

- Een initialisatiemethode `__init__` waaraan drie argumenten moeten doorgegeven worden: twee natuurlijke getallen a en b en een string met m karakters die gebruikt worden als de symbolen van het alfabet. Indien niet alle karakters van de gegeven string verschillend zijn, dan moet de methode een `AssertionError` opwerpen met de boodschap `alfabet bevat herhaalde symbolen`. Indien a en m niet copriem zijn, dan moet de methode een `AssertionError` opwerpen met de boodschap *a* en *m* zijn niet copriem. Hierbij moeten de cursieve fragmenten uiteraard ingevuld worden met de waarden van a en m. We geven nog mee dat je gebruik kan maken van de functie `gcd` uit de module `fractions` om de grootste gemene deler van twee gehele getallen te bepalen.
- Een methode `codeerSymbool` waaraan een string moet doorgegeven worden die bestaat uit één enkel karakter. Indien het karakter voorkomt in het alfabet van symbolen, dan moet de methode het gecodeerde symbool teruggeven. Anders moet de methode het gegeven karakter zelf teruggeven.
- Een methode `decodeerSymbool` waaraan een string moet doorgegeven worden die bestaat uit één enkel karakter. Indien het karakter voorkomt in het alfabet van symbolen, dan moet de methode het gedecodeerde symbool teruggeven. Anders moet de methode het gegeven karakter zelf teruggeven.
- Een methode `codeer` waaraan een string moet doorgegeven worden. De methode moet de gegeven string teruggeven waarin alle karakters die voorkomen in het alfabet van symbolen vervangen werden door hun gecodeerde symbool. Alle karakters die niet voorkomen in het alfabet van symbolen moeten behouden blijven.
- Een methode `decodeer` waaraan een string moet doorgegeven worden. De methode moet de gegeven string teruggeven waarin alle karakters die voorkomen in het alfabet van symbolen vervangen werden door hun gedecodeerde symbool. Alle karakters die niet voorkomen in het alfabet van symbolen moeten behouden blijven.

Bovendien moet het mogelijk zijn om twee objecten van de klasse `T52` bij elkaar op te tellen aan de hand van de `+` operator. Het resultaat moet opnieuw een object zijn van de klasse `T52`, dat een T52 *Geheimschreiber* machine voorstelt die hetzelfde resultaat oplevert dan wanneer we eerst berichten zouden coderen met de machine links van de `+` operator en die daarna nog eens zouden coderen met de machine rechts van de `+` operator. Indien de twee `T52` objecten die bij

elkaar opgeteld worden niet hetzelfde alfabet gebruiken, dan moet een AssertionError opgeworpen worden met de boodschap alfabetten zijn verschillend.

Voorbeeld

```
>>> machine1 = T52(3, 5, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')

>>> machine1.codeerSymbool('G')
'X'
>>> machine1.codeerSymbool('S')
'H'
>>> machine1.codeerSymbool('-')
'-'

>>> machine1.decodeerSymbool('X')
'G'
>>> machine1.decodeerSymbool('H')
'S'
>>> machine1.decodeerSymbool('-')
'-'

>>> machine1.codeer('G-SCHREIBER')
'X-HLAERDIRE'
>>> machine1.decodeer('X-HLAERDIRE')
'G-SCHREIBER'

>>> machine2 = T52(17, 11, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
>>> machine2.codeer('X-HLAERDIRE')
'M-AQLBOKROB'

>>> machine12 = machine1 + machine2
>>> machine12.codeer('G-SCHREIBER')
'M-AQLBOKROB'

>>> T52(4, 5, 'ABCDEFGHIJKLMMLKJIHGFEDCBA')
Traceback (most recent call last):
AssertionError: alfabet bevat herhaalde symbolen

>>> T52(4, 5, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
Traceback (most recent call last):
AssertionError: 4 en 26 zijn niet copriem

>>> machine1 + T52(17, 11, 'abcdefghijklmnopqrstuvwxyz')
Traceback (most recent call last):
AssertionError: alfabetten zijn verschillend
```