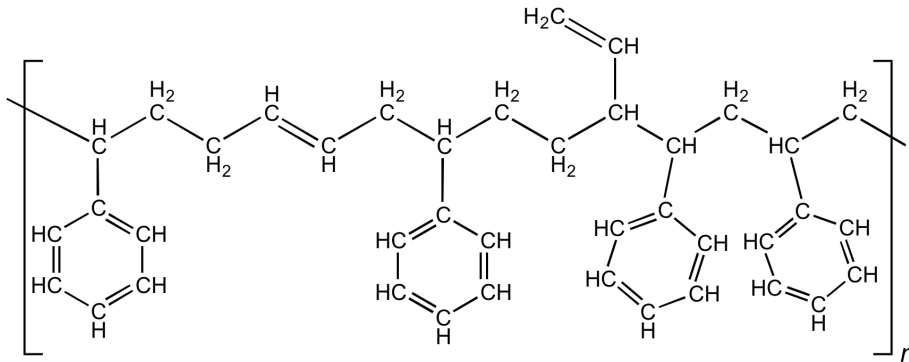


Periodic copolymers

When two or more different monomers unite together to polymerize, the result is called a **heteropolymer** or a **copolymer**. Polymers that contain only a single type of repeat unit are known as **homopolymers**. Commercially relevant copolymers include acrylonitrile butadiene styrene (ABS) plastic, styrene-butadiene rubber (SBR), nitrile rubber, styrene-acrylonitrile, styrene-isoprene-styrene (SIS) and ethylene-vinyl acetate.



Chemical structure of styrene-butadiene.

Since a copolymer consists of at least two types of constituent units (also **structural units**), they can be classified based on how these units are arranged along the chain. **Periodic copolymers**, for example, are copolymers whose structural units are arranged in a repeating sequence. If we represent structural units by uppercase letters, the shorthand notation for periodic copolymers reads as $(ABC)_n$. In this notation, ABC indicates the **period** and $n \in \mathbb{N}$ ($n \geq 2$) indicates that the periodic copolymer consists of n repetitions of the period.

Assignment

In this exercise, copolymers are represented as strings containing uppercase letters only. Each uppercase letter represents one of the structural units used as building blocks of the copolymer. Your task is to check whether or not a given copolymer is periodic. If this is the case, you also have to derive the shorthand notation for the periodic copolymer. This is done in the following way:

- Write a function `copolymer` that takes a single string. If the given string contains uppercase letters only, it already represents a copolymer and the function must return the string itself. However, if the string is formatted as $(p)_n$, it represents the shorthand notation of a periodic copolymer with period p repeated $n \in \mathbb{N}_0$ times. In this case, the function must return the periodic copolymer in full length.
- Write a function `is_periodic` that takes two strings. The first argument represents a copolymer and the second argument a period (which is a copolymer itself). The function must return a Boolean value that indicates whether or not the given copolymer (first argument) is composed as a set number of repetitions of the given period (second argument).
- Use the function `is_periodic` to write a function `period` that takes a string representing a copolymer. The function must return the shortest possible period of the copolymer. This period may be equal to the given copolymer itself, in case the copolymer is not composed as the repetition of a shorter period. The function also has an optional parameter `minimal_repetition` that takes an integer $n \in \mathbb{N}_0$ (default value: 1). In case the

given copolymer is not composed out of a period that is repeated at least n times, the function must return the empty string.

- Use the function `period` to write a function shorthand that takes a string representing a copolymer. In case the copolymer is periodic with a period that is repeated at least twice, the function must return the short notation of the copolymer formatted as $(p)_n$. In this notation, p represents the shortest possible period of the periodic copolymer, and n in $\mathbb{N}_{\geq 0}$ ($n \geq 2$) represents the number of repetitions of the period in the copolymer. In case the copolymer is not periodic or if the copolymer is not composed of at least two repetitions of a period, the function must return the given copolymer itself.

Example

```
>>> copolymer('(AB)_18')
'ABABABABABABABABABABABABABABABABABABABABABABABABAB'
>>> copolymer('(ABBA)_9')
'ABBAABBAABBAABBAABBAABBAABBAABBAABBAABBAABBAABBA'
>>> copolymer('(ABABBAAAAABBB)_3')
'ABABBAAAAABBBABABBAAAAABBBABABBAAAAABBB'
>>> copolymer('ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB')
'ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB'

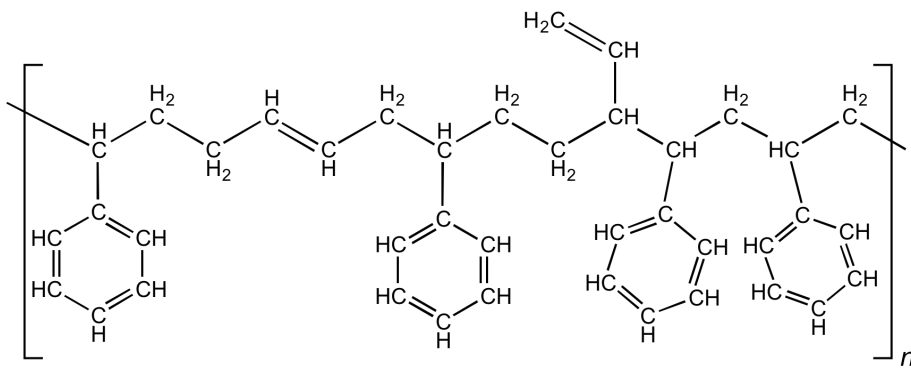
>>> is_periodic('ABABABABABABABABABABABABABABABABABABABABABABABABAB', 'AB')
True
>>> is_periodic('ABABABABABABABABABABABABABABABABABABABABABABABABAB', 'ABA')
False
>>> is_periodic('ABABABABABABABABABABABABABABABABABABABABABABABABAB', 'ABAB')
True
>>> is_periodic('ABABBAAAAABBBABABBAAAAABBBABABBAAAAABBB', 'ABABBAAAAABBB')
True

>>> period('ABABABABABABABABABABABABABABABABABABABABABABABABAB')
'AB'
>>> period('ABABABABABABABABABABABABABABABABABABABABABABABABAB', minimal_repetition=10)
'AB'
>>> period('ABABABABABABABABABABABABABABABABABABABABABABABABAB', 20)
""
>>> period('ABBAABBAABBAABBAABBAABBAABBAABBAABBAABBAABBAABBA')
'ABBA'
>>> period('ABABBAAAAABBBABABBAAAAABBBABABBAAAAABBB')
'ABABBAAAAABBB'
>>> period('ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB')
'ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB'
>>> period('ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB', 2)
""

>>> shorthand('ABABABABABABABABABABABABABABABABABABABABABABABABAB')
'(AB)_18'
>>> shorthand('ABBAABBAABBAABBAABBAABBAABBAABBAABBAABBAABBAABBA')
'(ABBA)_9'
>>> shorthand('ABABBAAAAABBBABABBAAAAABBBABABBAAAAABBB')
'(ABABBAAAAABBB)_3'
>>> shorthand('ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB')
'ABABBAAAAABBBABABBAAAAABBBBABBBAAAAABBB'
```

Heteropolymeren of **copolymeren** zijn polymeren die opgebouwd zijn uit verschillende monomeren. Dit in tegenstelling tot **homopolymeren** die uit één enkel monomeer zijn

opgebouwd. Onder de commercieel beschikbare copolymeren vinden we acrylonitril-butadieen-styreen (ABS) plastic, styreen-butadieen rubber (SBR), nitril-butadieen rubber en ethyleenvinylacetaat.



Chemische structuur van styrene-butadiene.

Omdat copolymeren bestaan uit minstens twee soorten samenstellende eenheden (**structureenheden**), kunnen ze ingedeeld worden volgens de manier waarop deze eenheden gerangschikt zijn. **Periodieke copolymeren** zijn bijvoorbeeld copolymeren waarbij de structureenheden gerangschikt zijn volgens een repeterende reeks. Als de structureenheden aangeduid worden door hoofdletters, dan kunnen periodieke copolymeren in verkorte notatie geschreven worden als $(ABC)_n$. Hierbij stelt ABC de **periode** voor, en geeft $n \in \mathbb{N}$ ($n \geq 2$) aan dat het periodiek copolymeer bestaat uit n herhalingen van de periode.

Opgave

In deze opgave worden copolymeren voorgesteld als strings die bestaan uit hoofdletters. Elke hoofdletter staat daarbij voor één van de structureenheden waaruit het copolymeer is opgebouwd. Je opdracht bestaat erin om voor een gegeven copolymeer na te gaan of het periodiek is. Indien dat het geval is, dan moet je ook de verkorte notatie van het copolymeer opstellen. Hiervoor ga je als volgt te werk:

- Schrijf een functie `copolymeer` waaraan een string moet doorgegeven worden. Indien de gegeven string enkel bestaat uit hoofdletters, dan stelt deze een copolymeer voor en moet de functie de string zelf als resultaat teruggeven. Indien de gegeven string de vorm $(p)_n$ heeft, dan stelt deze de verkorte notatie van een periodiek copolymeer voor met een periode p die $n \in \mathbb{N}_0$ keer herhaald wordt. De functie moet dan het volledig uitgeschreven periodiek copolymeer als resultaat teruggeven.
- Schrijf een functie `is_periodiek` waaraan twee strings moeten doorgegeven worden. Het eerste argument stelt een copolymeer voor en het tweede argument een periode (zelf ook een copolymeer). De functie moet een Booleaanse waarde teruggeven, die aangeeft of het gegeven copolymeer (eerste argument) bestaat uit een geheel aantal herhalingen van de gegeven periode (tweede argument).
- Gebruik de functie `is_periodiek` om een functie `periode` te schrijven waaraan een copolymeer moet doorgegeven worden. De functie moet de kortst mogelijke periode teruggeven. Deze periode kan eventueel gelijk zijn aan het gegeven polymeer zelf, indien het niet bestaat uit een herhaling van een kortere periode. De functie heeft ook nog een optionele parameter `minimale_herhaling` waaraan een getal $n \in \mathbb{N}_0$ kan doorgegeven worden (standaardwaarde: 1). Indien het gegeven copolymeer geen periode heeft die minstens n

keer herhaald wordt, dan moet de functie de lege string teruggeven.

- Gebruik de functie `periode` om een functie afkorting te schrijven waaraan een copolymeer moet doorgegeven worden. Indien het copolymeer periodiek is met een periode die minstens twee keer herhaald wordt, dan moet de functie de verkorte notatie van het copolymeer teruggeven onder de vorm $(p)_n$. Hierbij stelt p de kortst mogelijke periode van het periodiek copolymeer voor, en $n \in \mathbb{N}_0$ ($n \geq 2$) het aantal herhalingen van de periode in het copolymeer. Indien het copolymeer niet periodiek is of indien het copolymeer niet bestaat uit minstens twee herhalingen van een periode, dan moet de functie het copolymeer zelf teruggeven.

Voorbeeld

```
>>> copolymeer('(AB)_18')
'ABABABABABABABABABABABABABABABABAB'
>>> copolymeer('(ABBA)_9')
'ABBAABBAABBAABBAABBAABBAABBAABBA'
>>> copolymeer('(ABABBAAAABBB)_3')
'ABABBAAAABBBABABBAAAABBBABABBAAAABBB'
>>> copolymeer('(ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB)')
'ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB'

>>> is_periodiek('ABABABABABABABABABABABABABABABABAB', 'AB')
True
>>> is_periodiek('ABABABABABABABABABABABABABABABABAB', 'ABA')
False
>>> is_periodiek('ABABABABABABABABABABABABABABABABAB', 'ABAB')
True
>>> is_periodiek('ABABBAAAABBBABABBAAAABBBABABBAAAABBB', 'ABABBAAAABBB')
True

>>> periode('ABABABABABABABABABABABABABABABABAB')
'AB'
>>> periode('ABABABABABABABABABABABABABABABABAB', minimale_herhaling=10)
'AB'
>>> periode('ABABABABABABABABABABABABABABABABAB', 20)
''

>>> periode('ABBAABBAABBAABBAABBAABBAABBAABBAABBA')
'ABBA'
>>> periode('ABABBAAAABBBABABBAAAABBBABABBAAAABBB')
'ABABBAAAABBB'
>>> periode('ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB')
'ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB'
>>> periode('ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB', 2)
''

>>> afkorting('ABABABABABABABABABABABABABABABABAB')
'(AB)_18'
>>> afkorting('ABBAABBAABBAABBAABBAABBAABBAABBAABBA')
'(ABBA)_9'
>>> afkorting('ABABBAAAABBBABABBAAAABBBABABBAAAABBB')
'(ABABBAAAABBB)_3'
>>> afkorting('ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB')
'ABABBAAAABBBABABBAAAABBBBBABBBAAAABBB'
```