

Programowanie 1R – Ćwiczenia 4

Łańcuchy tekstowe (Stringi)

Mikołaj Marcinkowski

17 marca 2026

Zadania możesz wykonywać jako oddzielne skrypty `.py` lub w notatniku jupytera `.ipynb`, chyba że treść zadania wskazuje inaczej.

Zadania wzorowane na Serii 4 zadań autorstwa Bartłomieja Zglinickiego (<https://www.fuw.edu.pl/~bzglinicki/teaching/plr/zadania/seria-4/>), udostępnionej na licencji CC BY 4.0.

Wszystkie metody łańcuchów tekstowych wbudowane w Pythona są opisane w dokumentacji pod tym linkiem. Dodatkowo, biblioteka `string` zawiera kilka przydatnych napisów, np. `ascii_lowercase` – małe litery łacińskie, czy `punctuation` – znaki interpunkcyjne.

Zadanie 1. Palindrom

Palindromem nazywamy napis, który czytany od lewej do prawej i od prawej do lewej brzmi tak samo, jak "Anna", "kajak", "PKP", "Kobyła ma mały bok.". Definicję palindromu zwykle rozszerza się tak, że wielka i mała wersja danej litery są tożsame, a znaki interpunkcyjne i spacje ignoruje się przy sprawdzaniu.

Napisz funkcję, która przyjmuje napis i sprawdza, czy jest palindromem.

Zadanie 2. Szyfr Cezara

Szyfr Cezara z przesunięciem n polega na zastąpieniu liter szyfrowanej wiadomości literami znajdującymi się n pozycji dalej w alfabecie. Jeśli w szyfrowaniu przeskoczmy za literę z , zawijamy alfabet od początku. Na przykład dla $n = 3$, podstawiamy $a \mapsto d$, $b \mapsto e$, $\dots z \mapsto c$. Gdy $n < 0$, idziemy do tyłu. Małe litery są zastępowane małymi, a wielkie wielkimi. Pozostałe znaki nie ulegają zmianie.

Napisz funkcję, która przyjmuje wiadomość i przesunięcie n , po czym zwraca ją zaszyfrowaną szyfrem Cezara. Napisz też funkcję odszyfrowującą,

która dla danego przesunięcia cofa działanie szyfru Cezara. Dla uproszczenia możesz zignorować polskie litery.

Zadanie 3. Liczby rzymskie

Napisz funkcję, która przyjmuje liczbę całkowitą od 1 do 3999 włącznie i zwraca jej zapis używając cyfr rzymskich. Napisz też funkcję, która przyjmuje napis będący rzymskim zapisem liczby i zwraca jej wartość jako liczbę całkowitą. Druga funkcja powinna umieć przyjąć liczby zapisane zarówno wielkimi, jak i małymi literami.

Zadanie 4. Nawiasy

Ciąg znaków złożony z nawiasów $(,)$, $[,]$ i $\{, \}$ uważamy za zbalansowany, jeśli każdemu z nawiasów otwierających odpowiada właściwy nawias zamykający oraz jeśli nawiasy są poprawnie zagnieżdżone. Na przykład ciągi: $() []$, $() \{ [] \{ ([]) \}$ są zbalansowane, zaś ciągi $) ($, $[]$, $(\{ [])$ – nie.

- a) Napisz funkcję **sprawdz**, która przyjmuje napis składający się ze znaków $(,)$, $[,]$ i $\{, \}$ i sprawdza, czy jest zbalansowany.
- b) Wzorując się na poprzednim podpunkcie, napisz funkcję, która przyjmuje napis składający się ze znaków $(,)$, $[,]$ i $\{, \}$ i wypisuje informacje o każdej parze otwierający–zamykający nawias, który znalazła: rodzaj nawiasu i położenie nawiasu otwierającego i zamykającego (numery znaków w napisie). Skorzystaj z `enumerate`.
- c) Napisz funkcję **wypisz**, która przyjmuje liczbę naturalną n i zwraca listę (lub zbiór) wszystkich napisów długości $2n$ składających się wyłącznie z nawiasów $($ i $)$, które są zbalansowane.