

Programowanie I R

Zadania – seria 4.

Łańcuchy tekstowe.

Zadanie 1. palindrome – Palindromy.

Palindromem nazywamy wyrażenie brzmiące tak samo przy czytaniu od lewej strony do prawej, jak i odwrotnie. Przy badaniu, czy dane wyrażenie jest palindromem, nie należy brać pod uwagę wielkości liter ani znaków interpunkcyjnych. Palindromami są na przykład: „Anna”, „kajak”, „O, ty z Katowic, Iwo? Tak, Zyto!”

Napisz funkcję `is_palindrome`, która przyjmuje jako argument łańcuch tekstowy i zwraca wartość `True`, jeśli jest on palindromem, lub `False` w przeciwnym przypadku.

Korzystając z tej funkcji napisz program `palindrome`, który przyjmuje jako argumenty wywołania dowolną ilość łańcuchów tekstowych i wypisuje na standardowe wyjście te z nich, które są palindromami.

Zadanie 2. caesar – Szyfr Cezara.

Szyfr Cezara, nazywany też *szyfrem przesuwającym*, to jedna z najstarszych i zarazem najprostszych technik szyfrowania tekstu. Szyfrowanie łańcucha tekstowego szyfrem Cezara z przesunięciem $n \in \mathbb{Z}$ polega na zastąpieniu każdej z liter tego łańcucha literą występującą w alfabecie n pozycji za literą oryginalną, gdy $n \geq 0$, lub $|n|$ pozycji przed literą oryginalną, gdy $n < 0$. Przyjmujemy przy tym, że litery wielkie i małe tworzą odrębne zbiory (a więc litera wielka zawsze zostanie zastąpiona literą wielką, zaś litera mała – literą małą) oraz że zbiory te są uporządkowane cyklicznie (tzn. przed literą *A* występuje litera *Z*, zaś za literą *Z* – litera *A*; analogicznie dla liter małych). Znaki, które nie są literami (np. cyfry, znaki specjalne), nie są w żaden sposób zmieniane. Na przykład szyfrując łańcuch tekstowy *Programowanie13w@\$Pythonie* szyfrem Cezara z przesunięciem 3 dla 26-literowego alfabetu łacińskiego otrzymamy *Surjudprzdqlh13z@\$Sbwkrqlh*.

Napisz funkcję `caesar`, która przyjmuje jako argumenty liczbę całkowitą oraz łańcuch tekstowy. Funkcja ta powinna szyfrować przekazany jej jako drugi argument łańcuch za pomocą szyfru Cezara z przesunięciem zadanym liczbą całkowitą przekazaną jako pierwszy argument oraz zwracać zaszyfrowany łańcuch.

Korzystając z tej funkcji napisz program `caesar`, który

- jeśli pierwszym argumentem wywołania jest `encrypt`, szyfruje przekazany mu jako trzeci argument łańcuch tekstowy za pomocą szyfru Cezara z przesunięciem zadanym liczbą całkowitą przekazaną jako drugi argument, a następnie wypisuje zaszyfrowany łańcuch na standardowe wyjście,
- jeśli pierwszym argumentem wywołania jest `decrypt`, odszyfrowuje przekazany mu jako trzeci argument łańcuch tekstowy, zaszyfrowany za pomocą szyfru Cezara z przesunięciem zadanym liczbą całkowitą przekazaną jako drugi argument, a następnie wypisuje odszyfrowany łańcuch na standardowe wyjście.

Przyjmij, że litery pojawiające się w łańcuchach tekstowych należą do 26-literowego alfabetu łacińskiego.

Zadanie 3. brackets – Balansowanie ciągów nawiasów.

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.

Napisz program `brackets`, który:

- jeśli pierwszym argumentem wywołania jest `check` – sprawdza, czy ciąg nawiasów przekazany mu jako drugi argument wywołania jest zbalansowany i wypisuje na standardowe wyjście stosowną informację,
- jeśli pierwszym argumentem wywołania jest `fix` – wypisuje na standardowe wyjście liczbę nawiasów, które trzeba dopisać w ciągu złożonym wyłącznie z nawiasów okrągłych (i), przekazanym jako drugi argument wywołania, by ciąg ten był zbalansowany,
- jeśli pierwszym argumentem wywołania jest `list` – wypisuje na standardowe wyjście wszystkie zbalansowane ciągi nawiasów o długości $2n$, gdzie n jest liczbą całkowitą przekazaną jako drugi argument wywołania, złożone wyłącznie z nawiasów okrągłych (i).

Zadanie 4. roman – Liczby rzymskie.

Napisz program `roman`, który:

- jeśli pierwszym argumentem wywołania jest `r` – zamienia przekazaną mu jako drugi argument wywołania liczbę całkowitą zapisaną za pomocą cyfr arabskich na liczbę zapisaną za pomocą cyfr rzymskich i wypisuje ją na standardowe wyjście,
- jeśli pierwszym argumentem wywołania jest `a` – zamienia przekazaną mu jako drugi argument wywołania liczbę całkowitą zapisaną za pomocą cyfr rzymskich na liczbę zapisaną za pomocą cyfr arabskich i wypisuje ją na standardowe wyjście.

Opracowanie: Bartłomiej Zglinicki.