Laboratorium 5 — Ćwiczenia w wykorzystaniu kolekcji w języku Python.

Słowniki oraz zbiory

Zbiór (set) to kolekcja przechowująca jedynie unikatowe wartości, dodatkowo nie daje ona gwarancji, że dodane do niej wartości zachowają oryginalną kolejność.

Zamrożony zbiór (frozenset) kolekcja ta zachowuje się identycznie jak zwykły zbiór z tą różnicą, że po jej utworzeniu nie możemy jej modyfikować.

Słownik (dict) w przeciwieństwie do zbioru przechowuje pary wartości (klucz, wartość), gdzie pierwsza wartość musi być unikatowa. Podobnie jak w zbiorze słownik nie gwarantuje zachowania kolejności kluczy.

Zadanie 1 Utwórz napis o treści:

„Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Dolor sed viverra ipsum nunc aliquet bibendum enim. In massa tempor nec feugiat. Nunc aliquet bibendum enim facilisis gravida. Nisl nunc mi ipsum faucibus vitae aliquet nec ullamcorper. Amet luctus venenatis lectus magna fringilla. Volutpat maecenas volutpat blandit aliquam etiam erat velit scelerisque in. Egestas egestas fringilla phasellus faucibus scelerisque eleifend. Sagittis orci a scelerisque purus semper eget duis. Nulla pharetra diam sit amet nisl suscipit. Sed adipiscing diam donec adipiscing tristique risus nec feugiat in. Fusce ut placerat orci nulla. Pharetra vel turpis nunc eget lorem dolor. Tristique senectus et netus et malesuada."

Następnie znajdź unikatowe elementy. **Podpowiedź:** zamień występujące w nim duże litery na małe, usuń kropki oraz przecinki (dla każdej z tych operacji istnieje gotowa metoda), a następnie wykorzystując zbiór (**set**) znajdź unikatowe słowa. Ile słów zostało, a ile było oryginalnie?

Zadanie 2 Utwórz dwa zbiory:

$$A = \{1, 2, 3, 4, 5\},$$
$$B = \{2, 4, 5\},$$

a następnie korzystając z metod dostępnych dla zbiorów sprawdź, czy:

- A jest podzbiorem B,
- A jest nadzbiorem B,

oraz wyznacz:

- przecięcie A i B,
- sumę zbiorów A i B,
- różnicę zbiorów A i B,
- różnicę symetryczną zbiorów A i B.

Zadanie 3 Wyznacz iloczyn kartezjański powyższych zbiorów.

Zadanie 4 Wykorzystując słownik napisz skrypt, który zapyta o liczbę z zakresu $< 0, 99 >$, a następnie wyświetli jej słowną reprezentację. Słownik nie powinien zawierać wszystkich kombinacji liczb (**Podpowiedź**: w przypadku większości liczb dwucyfrowych możemy rozbić liczbę na dziesiątki oraz jedności w celu konwersji).

Zadanie 5 Zaimplementuj kodowanie Base64 działające dla napisów ASCII, wykorzystujące słownik do mapowania ciągu bitów na znak. Algorytm Base64 przypisuje każdemu 6bitowemu fragmentowi wejścia jeden z 64 wybranych symboli. Ponieważ pojedynczy znak ASCI posiada 8bitów, algorytm powinien analizować po 3 znaki naraz (24bity) i kodować je przy użyciu 4 znaków ($6\text{bit} \cdot 4 = 24\text{bity}$) z tabeli Base64. W przypadku gdy długość napisu nie jest podzielna przez 3, do ciągu bitów dopisuje się tyle zer, aby ciąg był podzielny przez 6, a za każde 6 bitów brakujących do 24 wstawia się znak dopełnienia „=".

Przykład dla słowa TEST:

Krok 1: W pierwszym kroku analizie podane zostaną 3 pierwsze znaki „TES" zamieniając każdą literę na ciąg bitów (polecenia ord oraz bin) zgodnie z tabelą ASCII otrzymamy „0101 0100 0100 0101 0101 0011". Dzieląc uzyskany ciąg na fragmenty po 6 bitów uzyskamy „010101", „000100", „010101", „010011". Jeśli podczas implementacji tablicy Base64 jako klucze użyliśmy liczb całkowitych, kolejnym krokiem będzie konwersja z systemu binarnego na dziesiętny: „21, 4, 21, 19". Pozostała zamiana otrzymanych wartości na konkretne litery zgodnie z tabelą Base64 — „VEVT".

Krok 2: Ponieważ w słowie TEST pozostała nam już tylko jedna litera na ostatnim etapie poddana analize zostanie tylko wartość „T", zamieniając ją do postaci binarnej uzyskamy „0101 0100". Jak widzimy daje nam to jeden ciąg 6bitowy i jeden 2bitowy. W związku z tym musimy uzupełnić drugi ciąg czterema zerami otrzymując „010101" oraz „000000", co daje litery „VA". Dwie brakujące litery zastępujmy znakami dopełnienia „=" (zakodowany tekst ma zawsze długość podzielną przez 4).

Wynik: Base64('TEST') ⇒ 'VEVTVA=='