

Warunki

1. W implementacjach można korzystać tylko z elementarnych konstrukcji Python'a (funkcje, instrukcje warunkowe, pętle, `range`, klasy użyte do definiowania struktur danych). **Nie wolno korzystać ze słowników i zbiorów, itp. Wolno korzystać z:**
 - (a) wbudowanego sortowania,
 - (b) wbudowanej kolejki priorytetowej (`PriorityQueue` z biblioteki `queue`).
2. Rozwiązania muszą być efektywne obliczeniowo (także w zadaniach, w których nie podajemy wprost ograniczenia na złożoność obliczeniową). Zadania o zbyt wysokiej złożoności będą otrzymywały obniżone oceny (lub 0).
3. Rozwiązania zadań proszę umieszczać w załączonych plikach (`zadX.py`).

Zadanie 1 (kody Huffmana)

Dana jest tablica symboli S oraz tablica ich częstości F . Przykładowe dane to:

```
S = ["a", "b", "c", "d", "e", "f" ]
F = [10, 11, 7, 13, 1, 20 ]
# symbol "a" ma częstość 10, symbol "b" ma częstość 11 itd.
```

Proszę napisać funkcję `huffman(S, F)`, która oblicza kod Huffmana dla takich danych, wypisuje każdy symbol (w kolejności z tablicy S) wraz z jego kodem, oraz łączną liczbę bitów potrzebną do wypisania napisu, w którym każdy symbol występuje podaną częstość razy. Przykładowy wynik dla podanego powyżej wejścia to:

```
a : 011
b : 10
c : 0101
d : 11
e : 0100
f : 00
dlugosc napisu: 150
```