**Лабораторна робота №4**

**Робота зі словами тексту**

Завдання:

* Використовуючи леми, отримані для тексту англійською мовою з лабораторної роботи №3, побудувати словник документу (якщо в лр№3 леми не було визначено, то можна використовувати стеми).
* Сформувати OHE слів документу та побудувати вектор мішку слів.
* Сформувати звіт заступного змісту:
* вхідний текст;
* код програми;
* результати роботи програми та висновки по результатам.

**Роботу виконав:** Гапей М.Ю. ПД-31

**Вхідний текст:**

One-hot encoding is often used for indicating the state of a state machine. When using binary or Gray code, a decoder is needed to determine the state. A one-hot state machine, however, does not need a decoder as the state machine is in the nth state if and only if the nth bit is high.

A ring counter with 15 sequentially ordered states is an example of a state machine. A 'one-hot' implementation would have 15 flip flops chained in series with the Q output of each flip flop connected to the D input of the next and the D input of the first flip flop connected to the Q output of the 15th flip flop. The first flip flop in the chain represents the first state, the second represents the second state, and so on to the 15th flip flop which represents the last state. Upon reset of the state machine all of the flip flops are reset to '0' except the first in the chain which is set to '1'. The next clock edge arriving at the flip flops advances the one 'hot' bit to the second flip flop. The 'hot' bit advances in this way until the 15th state, after which the state machine returns to the first state.

An address decoder converts from binary or Gray code to one-hot representation. A priority encoder converts from one-hot representation to binary or Gray code.

**Код програми:**

import numpy as np  
import collections  
import re  
  
def bag\_of\_words(file\_name):  
 texts = [open(file\_name, 'r').read().replace('\n', '')]  
 bag = [collections.Counter(re.findall(r'\w+', txt)) for txt in texts]  
  
 return bag  
  
  
def one\_hot\_encode(file\_name):  
 text = open(file\_name, 'r').read().replace('\n', '').split()  
 unique\_words = np.unique(text)  
  
 mapping = {}  
 for x in range(len(unique\_words)):  
 mapping[unique\_words[x]] = x  
  
 ohe = []  
  
 for word in text:  
 arr = list(np.zeros(len(unique\_words), dtype=int))  
 arr[mapping[word]] = 1  
 ohe.append(arr)  
  
 return ohe  
  
  
print(bag\_of\_words('text\_en.txt'))  
print(one\_hot\_encode('text\_en.txt'))  
  
**Результати роботи програми:**



**Висновки:**

Було проведено аналі тексту та сформовано OHE слова з тексту, а також побудувано вектор мішку слів.