НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КРИПТОГРАФІЯ

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ №1**

Експериментальна оцінка ентропії на символ джерела відкритого текст

Виконали:

студентки групи ФБ-23

Гуз Вікторія

Шукалович Марія

**Мета роботи:** засвоєння понять ентропії на символ джерела та його надлишковості, вивчення та порівняння різних моделей джерела відкритого тексту для наближеного визначення ентропії, набуття практичних навичок щодо оцінки ентропії на символ джерела.

**Постановка задачі**

1. Написати програми для підрахунку частот букв і частот біграм в тексті, а також підрахунку 𝐻1 та 𝐻2 за безпосереднім означенням. Підрахувати частоти букв та біграм, а також значення 𝐻1 та 𝐻2 на довільно обраному тексті російською мовою достатньої довжини (щонайменше 1Мб), де імовірності замінити відповідними частотами. Також одержати значення 𝐻1 та 𝐻2 на тому ж тексті, в якому вилучено всі пробіли.

2. За допомогою програми CoolPinkProgram оцінити значення 𝐻 (10) , 𝐻 (20) , 𝐻 (30) .

3. Використовуючи отримані значення ентропії, оцінити надлишковість російської мови в різних моделях джерела.

**Хід роботи**

Для початку ми залишили у тексті тільки російські літери(очистили від англійських, цифр, розділових знаків і подвійних пробілів):

def clean\_text():

    with open('lab1.txt', 'r', encoding='utf-8') as file:

        content = file.read()

    content = content.lower()

    cleaned = re.sub(r'[^а-я\s]+', '', content)

    cleaned = re.sub(r'\s+', ' ', cleaned)

    cleaned = cleaned.strip()

    return cleaned

Також ми врахували, що моментами потрібно буде працювати з текстом без пробілів і очистили від них:

def remove\_spaces(text):

    return text.replace(" ", "")

З цим етапом труднощів не виникало.

Далі слід було обчислити частоту кожної літери, щоб використати ці дані у формулі обчислення ентропії Н1:

def letter\_frequencies(letter\_counts, total\_count):

    frequencies = {}

    for letter, count in letter\_counts.items():

        frequencies[letter] = count / total\_count

    sorted\_frequencies = dict(sorted(frequencies.items(), key=lambda item: item[1], reverse=True))

    return sorted\_frequencies

Також відсортували вивід частот у порядку спадання.

Н1 обраховували за наступною формулою:

p(i) – частота літери

n – кількість літер в алфавіті

Реалізація формули у коді:

def entropy\_H1(letter\_frequencies):

    entropy\_value = 0

    for p\_i in letter\_frequencies.values():

        if p\_i > 0:

            entropy\_value -= p\_i \* math.log2(p\_i)

    return entropy\_value

Також обчислили частоту біграм(з/без перетином(-у)), щоб використати у формулі обчислення ентропії Н2:

def bigram\_frequencies(bigram\_counts, total\_bigrams):

    frequencies = {}

    for bigram, count in bigram\_counts.items():

        frequencies[bigram] = count / total\_bigrams

    return frequencies

Формула для обчислення Н2:

p(i,j) – частота біграми

Реалізація формули у коді:

def entropy\_H2(bigram\_frequencies):

    entropy = 0

    for frequency in bigram\_frequencies.values():

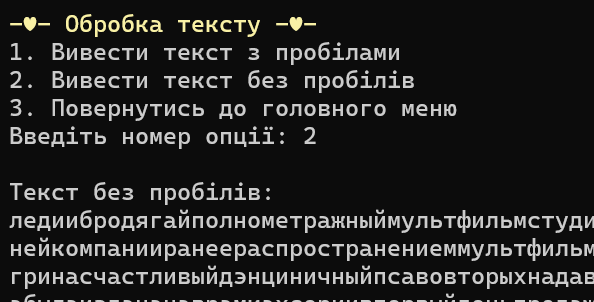
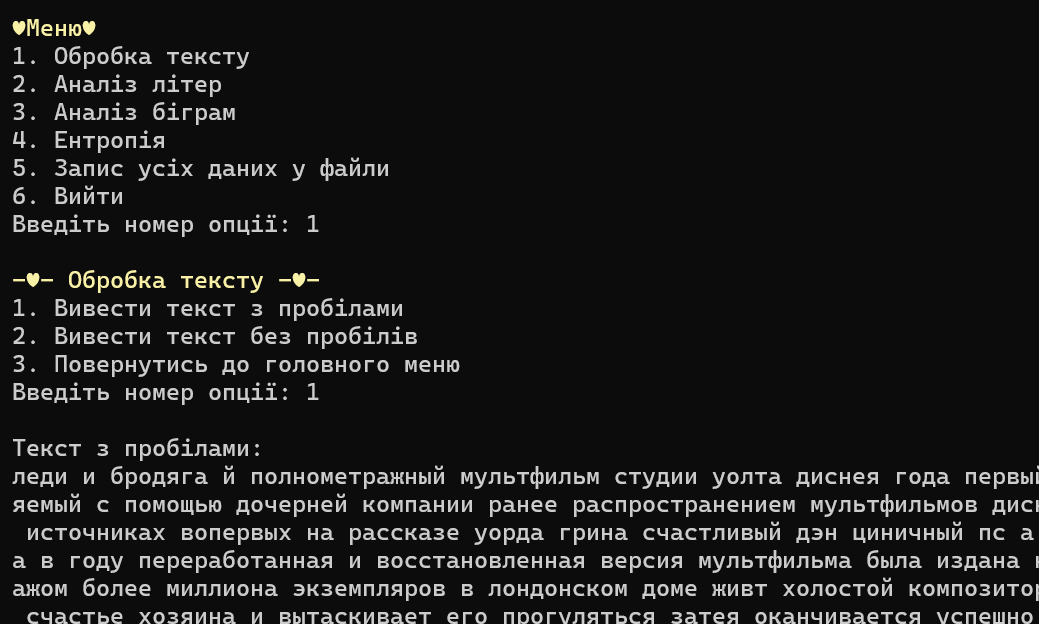
        if frequency > 0:

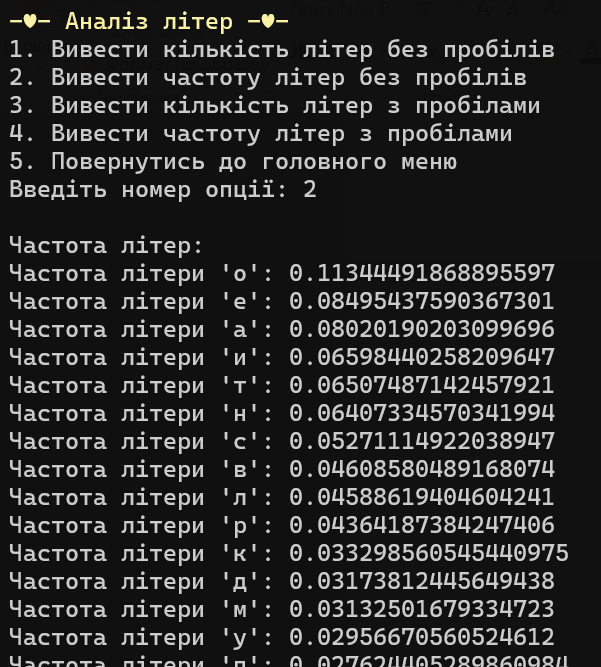
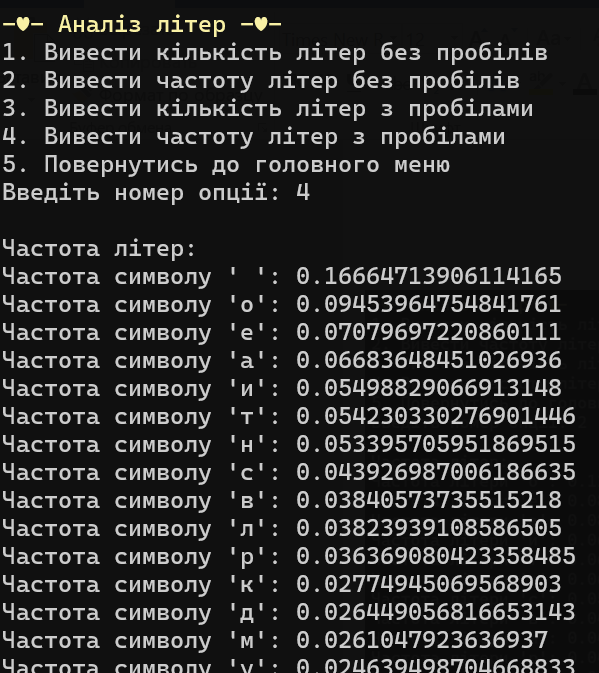
            entropy -= frequency \* math.log2(frequency)

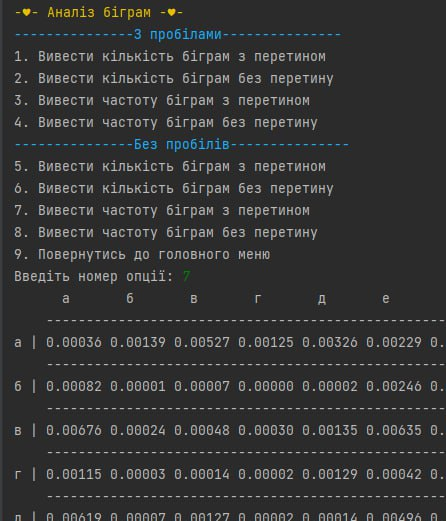
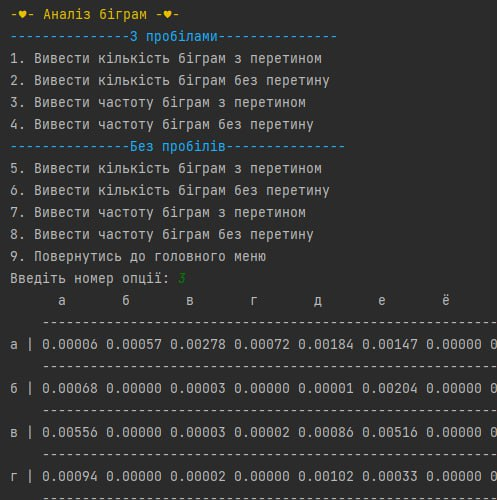
    return entropy / 2

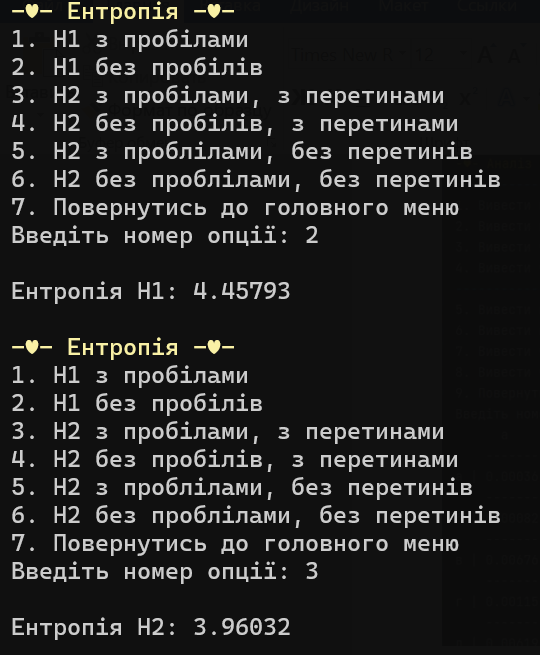
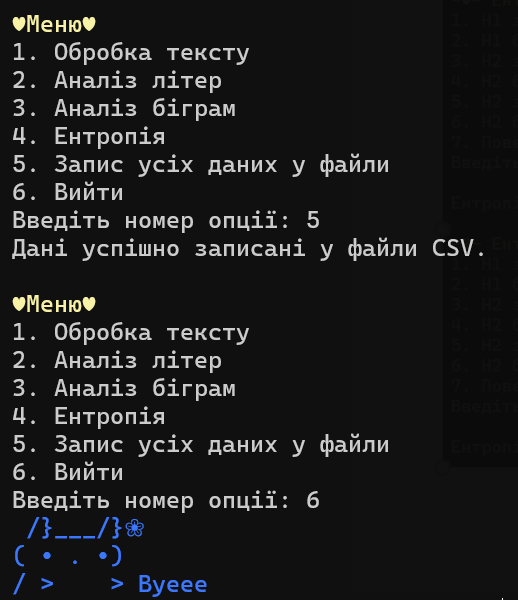
Під час обчислення ентропії проблем не виникло. Усі необхідні кроки виконано правильно, а результати відповідали очікуванням. Формули для обчислення ентропії використовувалися без помилок, дані оброблялися належним чином, і процес обчислення був успішним

Приклад виконання:



На наведених скриншотах лише частина даних для прикладу виконання, решта даних у прикріплених файлах:

* bigram\_counts\_no\_overlap\_with\_spaces.csv – кількість і частота біграм без перетину, з пробілами;
* bigram\_counts\_no\_overlap\_without\_spaces.csv – кількість і частота біграм без перетину, без пробілів;
* bigram\_counts\_with\_spaces.csv – кількість і частота біграм з перетином, з пробілами;
* bigram\_counts\_without\_spaces.csv - кількість і частота біграм з перетином, без пробілів;
* letter\_counts\_and\_frequencies\_with\_spaces.csv – кількість і частота літер з пробілами;
* letter\_counts\_and\_frequencies\_without\_spaces.csv – кількість і частота літер без пробілів;

До файлів не ввійшли біграми та літери з частотою 0, тому що неможливо взяти логарифм від 0.

Таблиця з отриманими значеннями ентропії:

|  |  |  |
| --- | --- | --- |
|  | Без пробілів | З пробілами |
| Н1 | 4.46598 | 4.37207 |
| Н2 з перетинами | 4.14440 | 3.96426 |
| Н2 без перетинів | 4.14597 | 3.96331 |

Таблиці з оцінками надлишковості російської мови:

Для тексту з пробілами 5*.*087 (33 літери і пробіл)

Для тексту без пробілів 5.044 (33 літери)

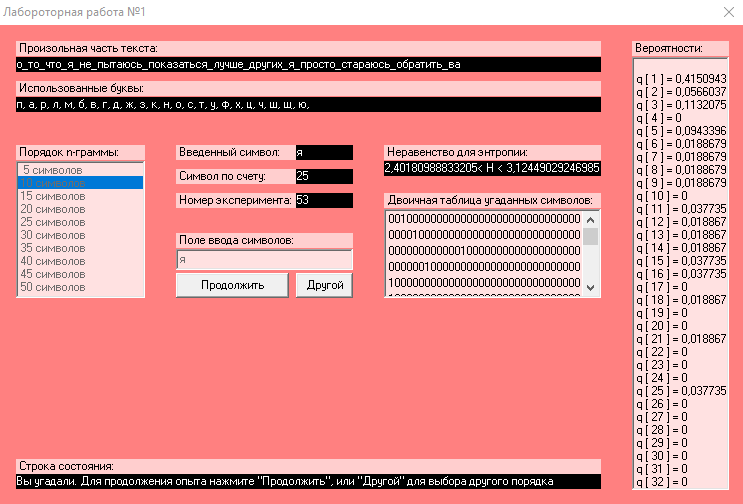
|  |  |  |
| --- | --- | --- |
|  | Без пробілів | З пробілами |
| Н1 | 0,11459 | 0,14054 |
| Н2 з перетинами | 0,17835 | 0,2207 |
| Н2 без перетинів | 0,17803 | 0,22089 |

Для тексту, який використовує CoolPinkProgram, 5 (32 літери)

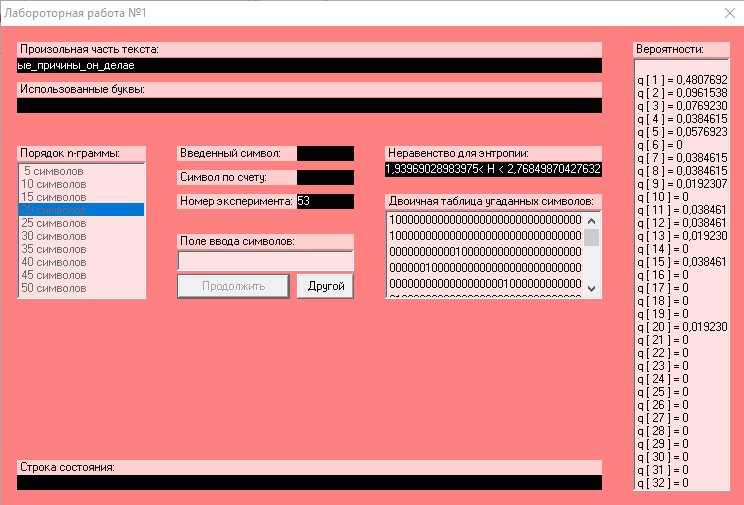
|  |  |  |
| --- | --- | --- |
|  | Надлишковість |  |
| 2,401809 <<3,12449 | 0,51963>>0,375101 |  |
| 1,93969<<2,76849 | 0,612062>>0,446302 |  |
| 1,18636<<1,783405 | 0,76272>>0,643319 |  |

**CoolPinkProgram**

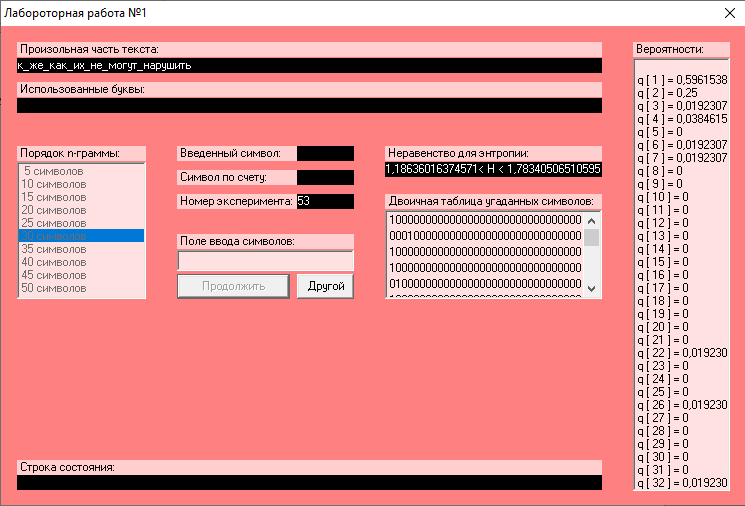
Для 10 символів:



Для 20 символів:



Для 30 символів:



**Висновки:** у ході виконання лабораторної роботи ми навчались визначати частоту літер та біграм в тексті, а також обраховувати значення його ентропії та надлишковості. Під час роботи з різними текстами ми помітили, що на ентропію суттєво впливають як довжина тексту, так і його структура (наявність повторюваних символів тощо). У текстах з високим рівнем впорядкованості (наприклад, з частими повтореннями тих самих слів чи символів) ентропія була нижчою, що вказує на меншу кількість інформації, необхідної для його передачі. Окрім цього, ми дослідили, що висока надлишковість свідчить про те, що текст містить багато передбачуваних елементів.