МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

Кафедра дискретного аналізу та інтелектуальних систем

**Звіт з лабораторної роботи №8**

**“Нормальні алгоритми маркова”**

Роботу виконав:

**Тимчишин Ярема Андрійович**

Студент групи Пмі-13

Перевірив:

**\_,**

\_ Львівського національного

університету імені Івана Франка

Львів – 2022

**ЗМІСТ**

ВСТУП 3

РОЗДІЛ 1 5

***1.1 Створення функцій для перевірки рядка, над яким далі будуть відбуватися дії*** 5

***1.2 Написання функції, яка буде повертати вже конвертований рядок*** 6

РОЗДІЛ 2 7

***2.1 Написання функції, яка буде перемножувати два унарних числа*** 7

РОЗДІЛ 3 10

***3.1 Створення тестів*** 10

***3.2 Результати тестів*** 11

ВИСНОВКИ 11

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 11

# ВСТУП

**Нормальні алгоритми Маркова**

Систему нормальних алгоритмів А. А. Марков запропонував у 1954 р. Алгоритми Маркова - це формальна математична система. Вони були основою для першої мови обробки рядів СОМІТ. Крім того, є подібність між моделлю Маркова і мовою СНОБОЛ, яка з'явилась після СОМІТ.

Загальна стратегія роботи алгоритму Маркова полягає у тому, щоб, застосувавши декілька операцій до вхідного рядка х, перетворити його у вихідний рядок у. Цей процес перетворення є звичайним у таких областях застосування ЕОМ, як редагування тексту або компіляція програми.

Простою продукцією (формулою підстановки) називають запис вигляду и -> w, де u,w - рядки в V \*, причому V не містить символів '->' та '.' . Величина и називається антицедентом, a w - консеквентом. Формула и -> w може бути застосована до рядка Z є V\*, якщо є хоча б одне входження и в Z. Інакше вона не застосовна до рядка Z. Якщо формула може бути застосована, то канонічне (перше зліва) входження и в Z заміняється на w. Наприклад, якщо формула 'ва' ->'с' може бути застосована до вхідного рядка 'ававав', то в результаті буде отримано рядок 'асава'. Водночас формула 'ваа'->'с' до рядка 'ававав' не може бути застосована.

Марківський алгоритм містить впорядковану множину продукцій Р1,Р2,...,Рn. Послідовність виконання алгоритму залежите від того, чи може бути застосована до рядка чергова формула підстановки. Виконання починається з перевірки першої продукції. Якщо вона може бути застосована до рядка, то рядок перетворюється. Якщо ж формула не може бути застосована (тобто в рядку не знайдено підрядка, який можна було б замінити), то відбувається перехід до перевірки наступної формули.

Марківський алгоритм завершується в одному з двох випадків.

1. До рядка не може бути застосована жодна з наявних формул підстановки.
2. До рядка застосовується заключна (термінальна) формула. Заключну підстановку позначають так:

**х ->. у** або **х->у.** , де **х, у є V\***

Розглянемо приклад. Нехай над словами з алфавіту \-{а,Ь,с} задано алгоритм з формулами підстановки

***P1: ’ab’ -> ’b’***

***P2: ’ac’ -> ’c’***

***P3: ’aa’ -> ’a’***

Цей алгоритм вилучає всі входження символа 'a' у рядку за винятком випадку, коли 'а' знаходиться у кінці рядка.

Простежимо роботу алгоритму, якщо вхідний рядок має вигляд **'bacaabaa'**. Далі символ => буде використовуватись для того, щоб вказати на результат перетворення, а підрядок, який підлягає заміні, будемо підкреслювати.

**‘baca abaa’** => застосовано продукцію P1

**‘bacabaa’** => застосовано продукцію P1

**‘bacbaa’** => застосовано продукцію P2

**‘bcbaa’** => застосовано продукцію P3

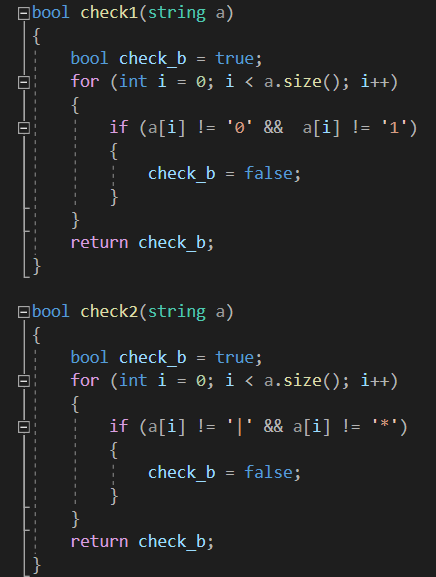
**‘bcba’**

Оскільки далі жодна з формул не може бути застосована, то на цьому робота алгоритму завершується.

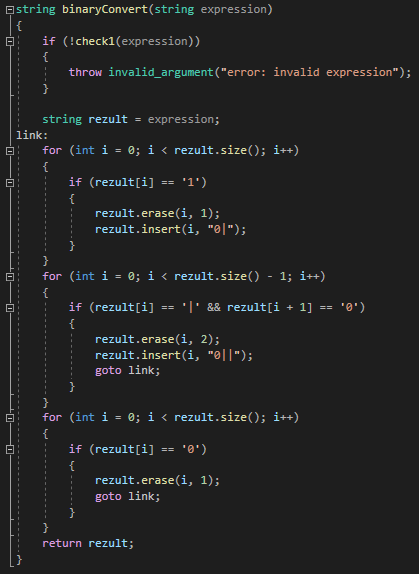
# РОЗДІЛ 1

**алгоритм, що перетворює двійкові числа в унарні**

## *1.1 Створення функцій для перевірки рядка, над яким далі будуть відбуватися дії*



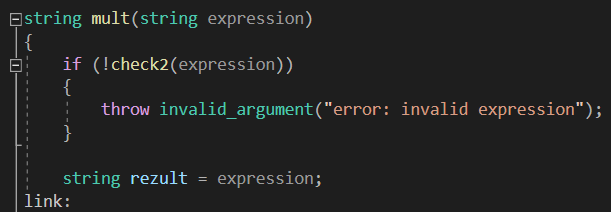
## *1.2 Написання функції, яка буде повертати вже конвертований рядок*

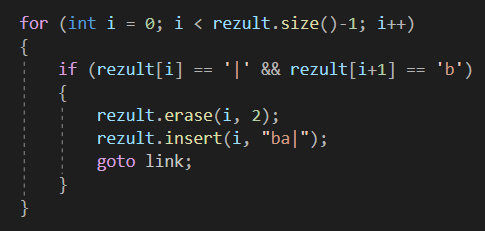


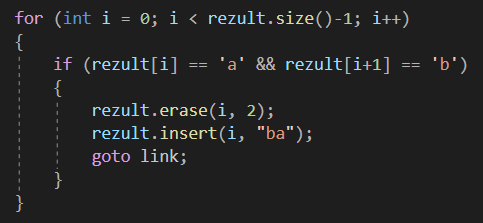
# РОЗДІЛ 2

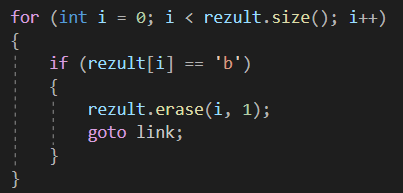
**алгоритм, що множить два унарних числа**

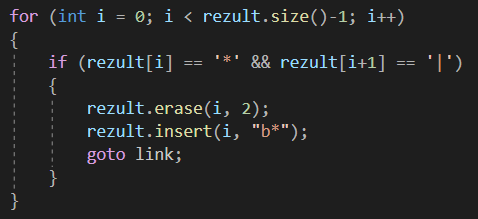
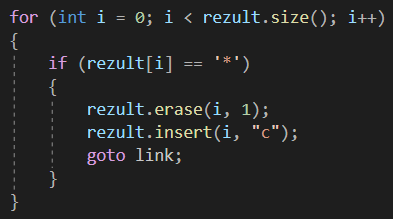
## *2.1 Написання функції, яка буде перемножувати два унарних числа*

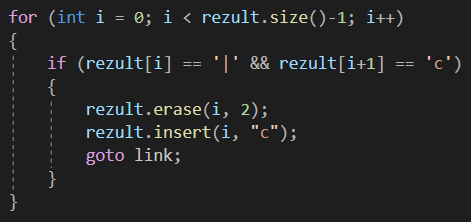


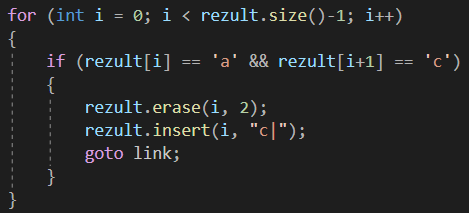


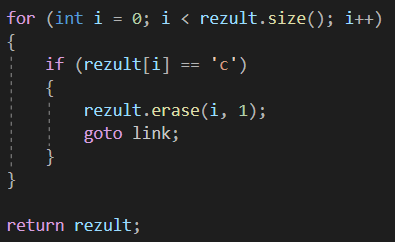






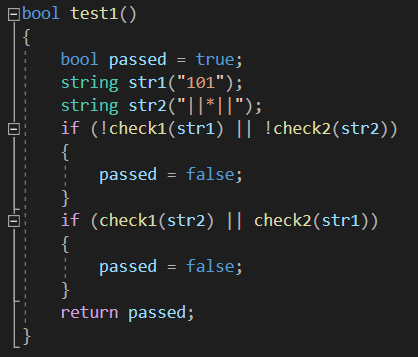


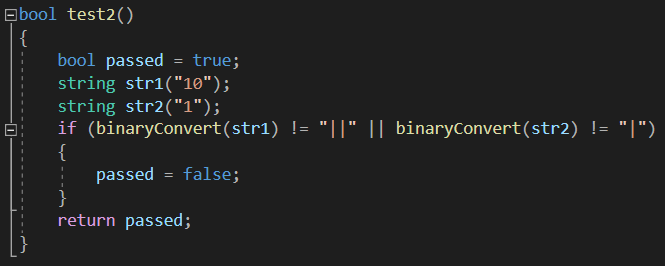


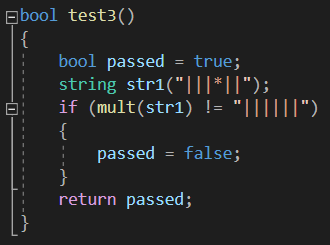
# РОЗДІЛ 3

**Перевірка справності написаних функцій**

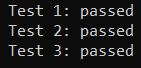
## *3.1 Створення тестів*







## *3.2 Результати тестів*



# ВИСНОВКИ

У результаті виконання роботи:

1. Ознайомився з нормальними алгоритмами Маркова.
2. Написав програму для реалізації двох алгоритмів Маркова.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. Костів. Структури даних: Навч. посібн. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2005. – 146 с