**Лабораторна робота 1 (до кінця 4 тижні навчання, 10б/6б)**

**Базові рядкові алгоритми**

Мова виконанняC або C++.

Нехай маємо текстовий файл, на який не накладаються обмеження щодо його розміру та довжини рядка в цьому файлі.

Текст складається зі слів, наприклад, ідентифікатори англійської мови. Слова перемежовуються проміжками, дужками, кодами операцій, взагалі символами, що природньо відділяють слово одне від одного. Правопис слів не має значення. Довжина слова не перевищує 30 літер.

В поле результату потрібно вивести слова без повторень, що задовольняють певній умові.

**Варіанти**

1. Знайти всі слова максимальної довжини.
2. Для кожного слова, що зустрічається у файлі, підрахувати, скільки разів це слово повторюється.
3. Знайти всі слова, що складаються лише з голосних літер.
4. Знайти лише ті слова, кожне з яких складається з літер, які не повторюються.
5. Знайти лише ті слова, які мають подвоєні приголосні літери.
6. Знайти лише ті слова, які мають найбільші підланцюжки приголосних літер.
7. Знайти ті слова, які в даному тексті зустрічаються найбільшу кількість разів.
8. Знайти ті слова, що мають більше голосних літер, ніж приголосних.
9. Знайти ті слова, які мають найбільшу кількість різних літер (тобто літери, які повторюються у слові, не враховувати).
10. Впорядкувати слова за їхньою довжиною.
11. Впорядкувати слова у відповідності зі зростанням кількості голосних літер, що містяться в словах.
12. Впорядкувати слова у відповідності зі зростанням кількості приголосних літер, що містяться в словах.
13. Нехай відстань між двома словами – кількість позицій, що відрізняються буквами. Знайти всі пари слів із найбільшою відстанню.
14. \*\* Нехай відстань між двома словами – кількість позицій, що відрізняються буквами. Нехай *m* - найбільша з відстаней між всіма парами слів. Знайти всі максимальні за включенням множини слів, які містять лише ті слова, відстань між будь-якими двома з них є *m*.

**Оцінювання:**

5 балів - коректний розвʼязок (демонстрація тестів);

1. балів - відповіді на питання по коду;

-4 бали - мова не C або C++;

+2 бали - за 14 варіант (*узгоджується з викладачем індивідуально*);

за невчасну здачу (запізнення із завантаженням більше, ніж 1 тиждень) - 50% від отриманої оцінки

**При виявленні плагіату (суттєвих запозичень в коді) - оцінка 0 балів за всю роботу.**

**Лабораторна робота 2 (до кінця 7 тижня навчання, 10б/6б)**

**Скінченні автомати та дії з ними**

Мова виконанняC, C++.

Розробити та реалізувати представлення скінченного автомата в пам`яті ПК.

**Варіанти**

1. Встановити, які літери вхідного алфавіту не сприймає скінченний детермінований автомат
2. Виявити недосяжні та тупикові стани скінченного автомату.
3. Розробіть алгоритм та реалізуйте програму, що моделює роботу недетермінованого скінченного автомата.
4. Розробити та реалізувати алгоритм перевірки, чи допускає скінченний автомат слова виду *w=w*0*w*1, де *w*0 – наперед задане (фіксоване) слово, а *w*1 – деяке слово, таке що *w* допускається скінченним автоматом. На вхід алгоритму подавати *w*0.
5. Розробити та реалізувати алгоритм перевірки, чи допускає скінченний автомат слова виду *w=w*1*w*0, де *w*0 – наперед задане (фіксоване) слово, а *w*1 – деяке слово, таке що *w* допускається скінченним автоматом. На вхід алгоритму подавати *w*0.
6. Розробити та реалізувати алгоритм перевірки, чи допускає скінченний автомат слова виду *w=w*1*w*0*w*2, де *w*1 та *w*2 – наперед задані (фіксовані) слова , а *w*0 – деяке слово, таке що w допускається скінченним автоматом. На вхід алгоритму подавати *w*1 та *w*2.
7. Розробити та реалізувати алгоритм перевірки: чи допускає скінченний автомат слова виду *w=w*1*w*0*w*2, де *w*0 – наперед задане (фіксоване) слово, а *w*1 та *w*2 – деякі слова, такі що *w* допускається скінченним автоматом. На вхід алгоритму подавати *w*0.
8. \* Знайти всі слова (без періодичних фрагментів), що сприймаються заданим скінченним автоматом.
9. \*Виявити, чи допускає (сприймає) скінченний автомат всі можливі (виходячи з алфівіту) слова заданої довжини *k*.
10. \*\*Реалізувати алгоритм пошуку слова мінімальної довжини, що допускається двома скінченними автоматами.
11. \*\*Реалізувати алгоритм мінімізації детермінованого скінченного автомата.
12. \*\*Реалізувати алгоритм перетворення недетермінованого скінченного автомата (без *ε*-переходів) в еквівалентний йому детермінований скінченний автомат
13. \*\*\* Виявити, чи є еквівалентними два задані детерміновані скінченні автомати.
14. \*\*\*Виявити, чи допускає (сприймає) скінченний автомат (скінченні) слова довільної (будь-якої) довжини.
15. \*\*\*Виявити, чи допускає (сприймає) скінченний автомат (скінченні) слова довільної (будь-якої) непарної довжини.
16. \*\*\*\*Виявити, чи є еквівалентними два задані недетерміновані скінченні автомати.
17. \*\*\*\*Побудувати регулярний вираз, що задає мову, яка розпізнається скінченним автоматом (мову слів, що ним сприймаються).

**Оцінювання:**

1. балів - коректний розвʼязок (демонстрація тестів);

5 балів - відповіді на питання по коду;

-4 бали - мова не C, C++;

+від 1 до 4 балів - за 8-17 варіанти (*узгоджується з викладачем індивідуально*);

за невчасну здачу (запізнення із завантаженням більше, ніж 1 тиждень) - 50% від отриманої оцінки.

**При виявленні плагіату (суттєвих запозичень в коді) - оцінка 0 балів за всю роботу.**

**Лабораторна робота 3 (до кінця 10 тижня навчання, 15б/9б)**

**Лексичний аналізатор мови програмування на основі базових рядкових алгоритмів**

Мова виконанняC, C++.

Реалізувати лексичний аналізатор мови програмування. Для зберігання класів лексем організувати таблиці. Вивести вміст таблиць після оброблення тексту програми.

Розрізняти щонайменше такі класи лексем:

* числа (десяткові, з плаваючою крапкою, шістнадцяткові);
* рядкові та символьні константи;
* директиви препроцесора;
* коментарі;
* зарезервовані слова;
* оператори;
* розділові знаки;
* ідентифікатори.

Позначати ситуації з помилками. Наприклад, нерозпізнавані символи.

Можливі варіанти виведення результату роботи програми:

* «розфарбовування» тексту програми;
* виведення пар <лексема, тип лексеми>;
* виведення лексем, що зустрічались в програмі, розбитими на класи.

**Варіанти**

1. Мова програмування C.
2. Мова програмування C++.
3. Мова програмування Pascal.
4. Мова програмування C#.
5. Мова програмування Delphi.
6. Мова програмування FORTRAN.
7. Мова програмування Java.
8. Мова програмування Swift.
9. Мова програмування Kotlin.
10. Мова програмування Ruby.
11. Мова програмування Python.
12. Мова програмування Go.
13. Мова програмування Visual Basic.
14. Мова скриптів Java Script.
15. Мова скриптів PHP.
16. Мова скриптів ASP.

**Оцінювання:**

1. балів - коректний розвʼязок (демонстрація тестів);

5 балів - відповіді на питання по коду;

-6 балів - мова не C, C++;

+від 1 до 4 балів - за додаткові мови (*узгоджується з викладачем індивідуально*);

за невчасну здачу (запізнення із завантаженням більше, ніж 1 тиждень) - 50% від отриманої оцінки.

**При виявленні плагіату (суттєвих запозичень в коді) - оцінка 0 балів за всю роботу.**

**Лабораторна робота 4 (до кінця 13 тижня навчання; 15б/9б)**

**Лексичний аналізатор мови програмування на основі скінченних автоматів**

Мова виконанняC, C++.

Реалізувати лексичний аналізатор мови програмування. Для зберігання класів лексем організувати таблиці. Вивести вміст таблиць після оброблення тексту програми.

Розрізняти щонайменше такі класи лексем:

* числа (десяткові, з плаваючою крапкою, шістнадцяткові);
* рядкові та символьні константи;
* директиви препроцесора;
* коментарі;
* зарезервовані слова;
* оператори;
* розділові знаки;
* ідентифікатори.

Позначати ситуації з помилками. Наприклад, нерозпізнавані символи.

Можливі варіанти виведення результату роботи програми:

* «розфарбовування» тексту програми;
* виведення пар <лексема, тип лексеми>;
* виведення лексем, що зустрічались в програмі, розбитими на класи.

**Варіанти**

1. Мова програмування Go.
2. Мова програмування Python.
3. Мова програмування Ruby.
4. Мова програмування Kotlin.
5. Мова програмування Swift.
6. Мова програмування Visual Basic.
7. Мова програмування Java.
8. Мова програмування Delphi.
9. Мова програмування C#.
10. Мова програмування Pascal.
11. Мова програмування C++.
12. Мова програмування C
13. Мова програмування FORTRAN.
14. Мова скриптів Java Script.
15. Мова скриптів PHP.
16. Мова скриптів ASP.

**Оцінювання:**

1. балів - коректний розвʼязок (демонстрація тестів);

5 балів - відповіді на питання по коду;

-6 балів - мова не C, C++;

+від 1 до 4 балів - за додаткові мови (*узгоджується з викладачем індивідуально*);

за невчасну здачу (запізнення із завантаженням більше, ніж 1 тиждень) - 50% від отриманої оцінки.

**При виявленні плагіату (суттєвих запозичень в коді) - оцінка 0 балів за всю роботу.**