ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

За курсом «Інформатика і Програмування»

Студента групи МС-19-1

Політіки Владислава Дмитровича

Лабораторна робота №9.3

**Тема:** «Обробка текстової інформації»

Варіант 11

Завдання:

1. За допомогою текстового редактора створити файл, що містить текст. Довжина рядка тексту не повинна перевищувати 80 символів. Це вхідний файл.

2. Програма повинна

1) інформацію вхідного файлу записати у вихідний файл;

2) реалізувати дії, вказані в індивідуальному завданні і занести відповідну інформацію у вихідний файл.

3. Ім'я вхідного файлу задає користувач в командному рядку або під час виконання програми.

4. Ім'я вихідного файлу: Перші символи - не більше трьох символів з імені вхідного файлу, а решти символів \_out.

5. Вихідний файл повинен мати розширення .dat.

6. Програма повинна забезпечувати по запиту користувача:

1) виведення тексту вхідного файлу на екран дисплея;

2) виведення тексту вихідного файлу на екран дисплея.

7. Закінчені послідовності дій оформити у вигляді функцій. Всі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

8. Для вводу і виводу інформації (і для консолі, і для текстових файлів) застосувати засоби форматованого вводу/виводу.

9. Визначення функцій і головну функцію розмістити в двох окремих файлах.

Індивідуальне завдання (варіантом 11):

Визначити кількість цифр в вхідному файлі, замінити цифри відповідними словами.

**Про програму**

1. Опис розв’язку:

* **Опис функції main():**

З самого початку програма питає користувача ввести ім’я файлу, якщо ж воно не задане у командному рядку. Якщо кількість аргументів у командному рядку дорівнює двом (перший - .exe-файл(рішення цього коду), другий - задає користувач), то в програмі функція fopen() передає вказівник на початку файлу змінній fin, також за циклом у масив f\_name типу char передається другий аргумент командного рядка поелементно, поки не трапиться нульвой елемент. Проте за таким циклом сам нульовий елемент не перепишеться, тоді потрібно його присвоїти до саме того індексу на якому цикл зупинився. Якщо файл не зміг відкритися, то був отриманий нульовий вказівник. У варіанті, коли не зміг відкритися файл, що був вже заданий у командному рядку, або, якщо файл не був заданий (якщо кількість аргументів командного рядка більша за 2, то програма це сприймає як те, що файл заданий не був), у програмі має виконуватись певна умова, для її спрощення я при оголошенні об’єкту FILE вказівнику fin присвоїв нульовий вказівник. Тож ця умова - рівність fin з нульовим вказівником. За її виконанням входимо у цикл перевірки на відкриття вказаного файлу та такої умови, що кожен рядок відкритого файлу має не перевищувати 80 символів. Для виходу з циклу потрібне значення false, також потрібно, щоб виконувались переглянуті умови (файл повинен бути відкритий, рядок не перевищувати 80 символів), але вони мали значення false - значення виходу. У разі не виконання однієї з умов цикл повинен продовжуватись, тоді користуємось логічним оператором «або». У разі не виконання однієї з умов циклу повідомляється про те, яка саме умова не була виконана (тут звичайне розгалуження). Після вдалого відкриття файлу, програма, записавши назву у масив, шукає за циклом «поки не нульовий елемент з індексом, який перебирається від 0» індекс останнього елемента, далі за іншим циклом «від кінця до символу ‘\’» індекс елементу ‘\’, далі за можливістю дописує до масиву 3 елемента назви файлу (за циклом в якій є команда break в умові, коли трапилася крапка) та врешті-решт присвоює елементам з індексами від отриманих після останнього циклу значення ‘\_’, ‘o’, ‘u’, ‘t’, ‘.’, ‘d’, ‘a’, `t’, `\0’. Створюємо або відкриваємо з видаленням файл з отриманою назвою у режимі перезапису. Тут функція fopen() передає вказівник на початку файлу змінній fout. Користуючись функцією fgets(), що є умовою продовження циклу(тобто умова - можливість зчитування з файлу), отримаємо кожний рядок з заданого файлу, який виводиться на екран та записується у отриманий файл (користуємось функцією fprintf()). Для повторного зчитування заданого файлу закриваємо та відкриваємо файл. Далі, за допомогою циклу з умовою можливості зчитування (тут як і в попередньому разі користуємось функцією fgets()), передаємо кожний отриманий функцією gets() рядок функції convert\_our\_numbers\_and\_count\_amount\_of\_digits, що повертає у передані другий та третій аргументи (за адресою) результати своєї роботи. Також, після отримання “трансформованого” рядка, він виводиться на екран та дописується у певний файл. При підрахунку кількості цифр у вхідному файлі функція, якщо вона розглядає цифру з отриманого рядка, збільшує передане значення кількості вже знайдених цифр на одиницю. Після отримання всієї кількості цифр, що знахоходились у вхідному файлі, програма виводинь результат на екран та у вихідний файл.

* **Опис функції convert\_our\_numbers\_and\_count\_amount\_of\_digits():**

Це функція типу void, її параметри: 1 - char\*, 2 - char\*, 3 - int&. У перший передається вказівник на перший елемент рядка, у другий передається вказівник на перший елемент масиву, у який програма кладе результат зміни переданого рядка, у третій передається адрес змінної типу int за яким зберігається кількость вже знайдених цифр у файлі.

Функція за циклом розглядає всі елементи переданого рядка до нульвого включно (while (i <= strlen(inData))). Якщо розглянутий елемент не є розділювачем (перевіряє функція is\_let\_or\_num\_and\_amount\_of\_digits()), то символ записується у масив NextW. Зокрема, функція is\_let\_or\_num\_and\_amount\_of\_digits() перевіряє чи є даний символ цифрою, якщо так, то збільшує значення змінної за переданою адресою на одиницю. Якщо ж трапляється розділювач, то до масиву NextW дозаписується нульвий елемент, далі масив NextW перевіряється чи складають його елементи до нульового число (перевіряє функція is\_num()). Якщо це число, то масив передають функції конвертації (convert\_simple\_number()). Після конвертації, або якщо масив не пройшов перевірку на число, до нього дозаписують розділювач на якому функція на перевірку символа (is\_let\_or\_num\_and\_amount\_of\_digits()) отримала значення false(є розділювачем). Далі отриманий масив дозаписується до масиву, що утворить по виходу з функції конвертований рядок.

* **Опис функції is\_let\_or\_num\_and\_amount\_of\_digits():**

Ця функція отримує один символ та адресу розміщення змінної int, що зберігає кількість цифр. За умовою «символ==розділювач» функція повертає false (тобто символ - розділювач, або символ - не частина числа або слова), далі за умовою певного розташування символу у таблиці ASCII можна говорити чи це цифра чи ні (в програмі значення true умови if ('0' <= a && a <= '9') говорить, що символ є цифрою). Якщо трапляється цифра, то змінна за переданою адресою (що зберігає кількість цифр) збільшується на 1. Також при виході з умови «символ==розділювач» можна говорити, що символ - не розділювач (інакше б був вихід з функції), тож тоді повертається значення true.

* **Опис функції is\_num():**

Функція перевіряє чи складають елементи до нульового числа переданого масиву число.

Якщо перший елемент - нульовий елемент, тоді це не число (false). Якщо це не так - вихід не здійснюється, а тоді перевіряється за циклом всі елементи масиву до нульового не включно. За виконанням умови «символ, що розглядається - не є частиною числа»(if((fData[i] < '0' || fData[i] > '9') && !(i == 0 && fData[i] == '-'))) здійснюється вихід з функції, повертається значення false. Після виходу з циклу можна говорити, що масив(до нульового елемента) утворює число - повертається значення true.

* **Опис функції convert\_simple\_number():**

Передається вказівник на перший елемент масиву, що утворює число (до нульового елемента). Після виходу цей масив стає конвертованим (за завданням).

Розглядаються всі елементи переданого масиву до нульового (за циклом while (inData[i] != '\0')). Ряд конвертованих значень записується до масиву keeping. Між цими записами додається пробіл (за допомогою функції My\_strcat()). У разі виконання еквівалентності між елементом переданого масиву і певним символом дозаписується символ словами до масиву keeping(такої конвертації повребує завдання, дозаписування відбувається за допомогою функції My\_strcat()). Після виходу з циклу дописується нульовий елемент до масиву keeping. За допомогою функції (str\_copy()) цей масив копіюється в переданий масив.

Також можна зауважити, що між конвертованими символами розташовані лапки - це робтиться за допомогою звичайного присвоювання певним елементам з певними індексами лапок.

* **Опис функції My\_strcat():**

Передаються два вказівники на два різні масиви. Результат: дозапис до першого масиву другого.

Нехай A - перший масив, B - другий, Ai - індекс для позначення якогось елемента в масиві A, Bi - індекс для позначення якогось елемента в масиві B.

За циклом “поки не нульовий елемент у масиві A”(for (Ai = 0; A[Ai] != '\0'; Ai++)) знаходимо індекс нульового елемента(зберігається у змінній Ai). Далі за циклом “поки не нульовий елемент у масиві B”(for (Bi = 0; B[Bi] != '\0'; Bi++)) елементам, починаючи з індексу Ai, маємо записати масив B. Тож нехай розглядається деякий індекс Bi, він має бути скопійований у індекс Ai + Bi. Для цього на кожній ітерації здійснюється присвоєння(A[Ai + Bi] = B[Bi]), а після виходу з циклу має бути дозаписаний останній елемент(A[Ai + Bi] = '\0';), бо коли B[Bi] <=> ‘\0’, то здійснюється вихід з циклу.

* **Опис функції str\_copy:**

Передаються два вказівники на два різні масиви. Результат: копіювання другого масиву у перший.

Нехай A1 - перший масив, A2 - другий, i - необхідна змінна типу int.

Тут все просто: за циклом “поки не трапився нульовий елемент у масиві A2” присвоюємо елементам з масиву A1 елементи з масиву A2, що мають відповідні індекси до індексів з A1. Як і в функції (**My\_strcat()**) після виходу з циклу має бути скопійоманий останній елемент(нульовий), тож елементу A1[i] (i - індекс, що вказує на нульовий елемент після виходу з циклу) присвоюємо ‘\0’.

1. Вихідний текст програми розв’язку задачі:

Файл «Лабораторна\_робота\_9(3).cpp»:



Файл «Source.cpp»:

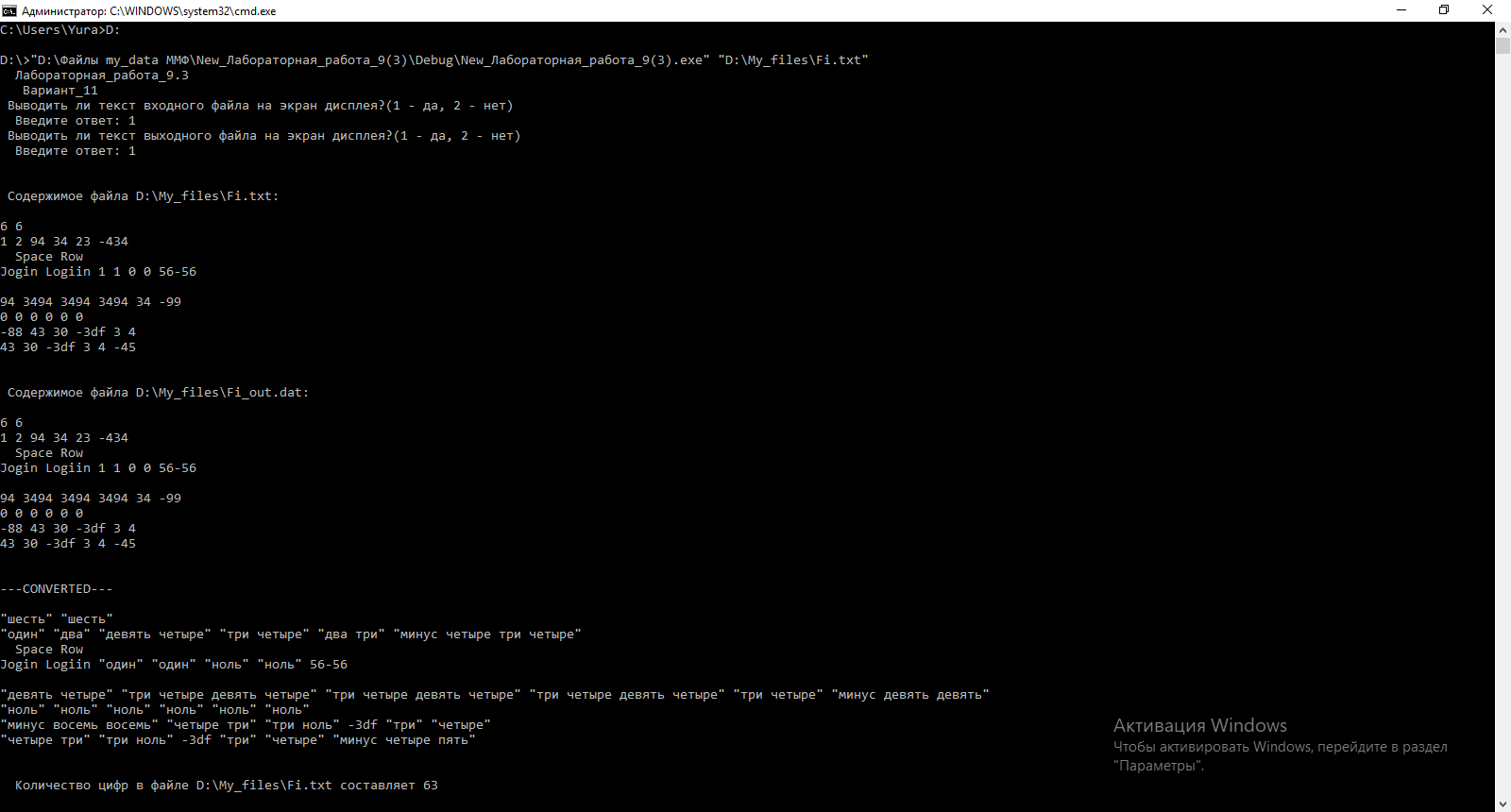


header-файл «Header.h»:



1. Опис інтерфейсу:

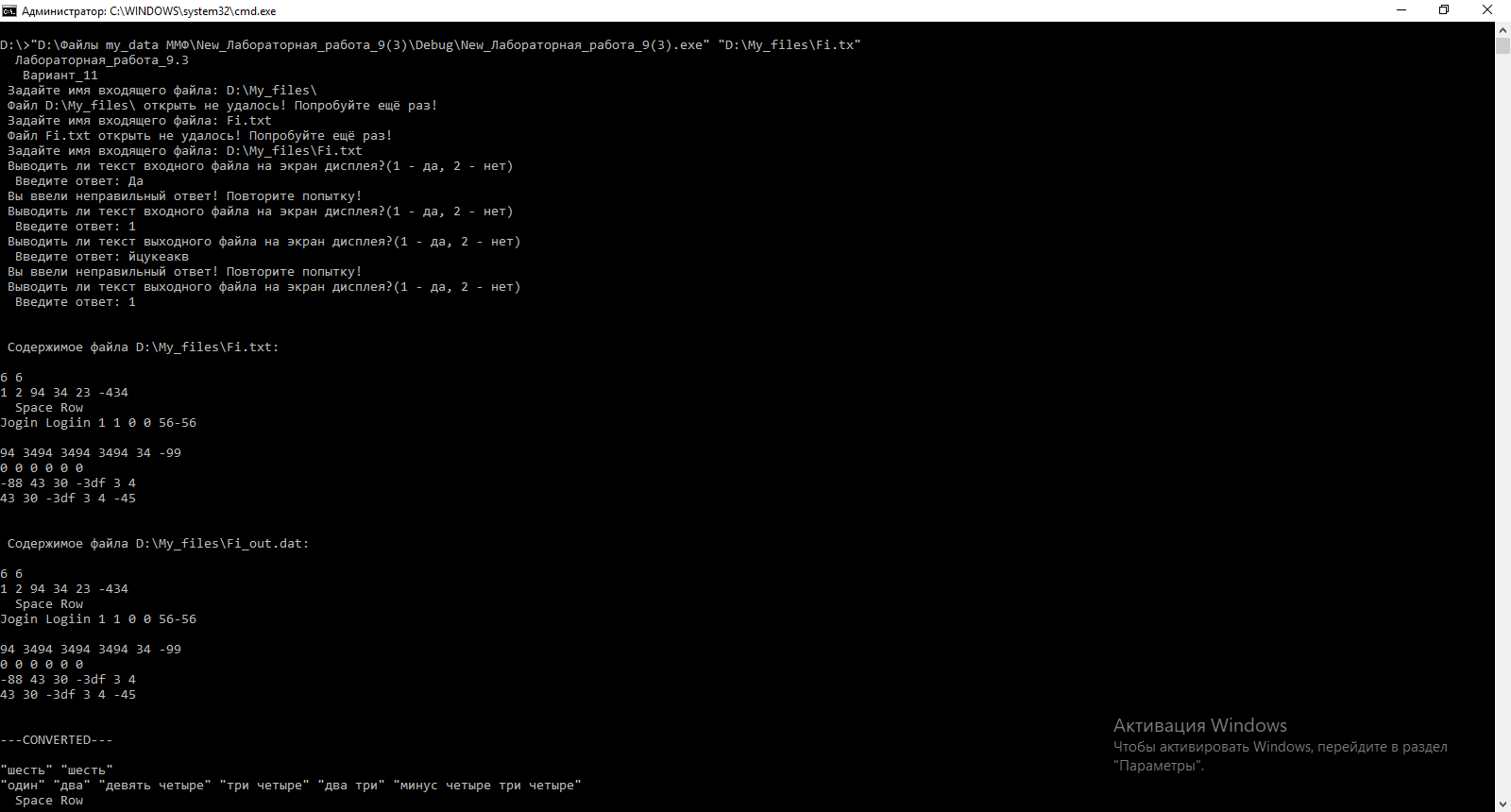
Робота програми при вводі назви файлу у командному рядку:



Також було створено файл Fi\_out.dat:



Робота програми у разі створення помилок:



1. Опис тестових прикладів:

При всіх моїх тестових прикладах, програма виконує свою роботу вірно.

1. Аналіз помилок:

Помилок не має.