**Лабораторна робота № 6.****Тема: Створення програм для обробки текстових даних**

МЕТА РОБОТИ: отримати практичні навики по опису та обробці текстової інформації в програмах на С++

**Варіант 14**

Задача №1

Умова:

. Ввести задане речення та вивести слова, у яких немає сполучення літер «про» та літери «р»:

«Профессионализм — это стабильность результатов».

1. Просимо користувача ввести речення з використанням **cout**.
2. Створення масиву символів **sentence** для зберігання введеного речення. Визначення константи **maxSentenceLength** для обмеження довжини речення.
3. Зчитування введеного речення користувача за допомогою **cin.getline** та збереження його у масив **sentence**.
4. Створення масиву символів **word** та змінної **wordLength** для обробки слів у введеному реченні.
5. Виведення повідомлення "Слова без 'про' та 'р':".
6. Перехід по символах у масиві **sentence** за допомогою циклу **for**.
7. Перевірка кожного символу на те, чи це пробіл або завершення рядка (символ '\0').
8. Якщо символ - пробіл або завершення рядка, то поточне слово завершується символом '\0'. Потім перевіряється, чи містить це слово підрядок "про" або букву "р". Для цього використовується внутрішній цикл.
9. Якщо слово містить "про" або "р", змінна **has\_pro\_or\_r** встановлюється в **true**, і це слово не виводиться. В іншому випадку слово виводиться на екран.
10. Після обробки слова, змінна **wordLength** обнуляється для подальшого зчитування наступного слова.
11. Завершення програми.

Код програми:

#include <iostream>

#include <Windows.h>

#include <cstring>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Введіть речення: ";

const int maxSentenceLength = 1000;

char sentence[maxSentenceLength];

cin.getline(sentence, maxSentenceLength);

char word[maxSentenceLength];

int wordLength = 0;

cout << "Слова без 'про' та 'р': ";

for (int i = 0; i <= strlen(sentence); i++) {

char c = sentence[i];

if (c == ' ' || c == '\0') {

word[wordLength] = '\0';

bool has\_pro\_or\_r = false;

for (int j = 0; j < wordLength; j++) {

if (word[j] == 'п' && (strstr(word, "про") != nullptr)) {

has\_pro\_or\_r = true;

break;

}

else if (word[j] == 'р') {

has\_pro\_or\_r = true;

break;

}

}

if (!has\_pro\_or\_r) {

cout << word << " ";

}

wordLength = 0;

}

else {

word[wordLength++] = c;

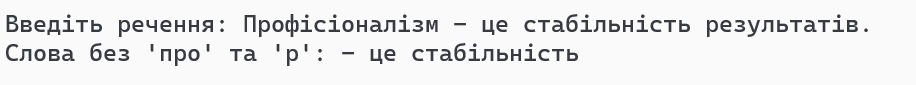
}

}

cout << endl;

}

Результат:



Задача №2

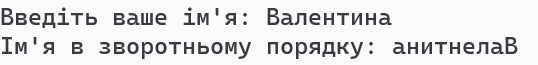
**Умова:**

Вивести ім’я в зворотньому порядку.

Алгоритм:

1. Створення масиву символів **name** розміром 100 для зберігання введеного імені користувача.
2. Виведення повідомлення "Введіть ваше ім'я:" з використанням **cout**.
3. Зчитування введеного імені користувача за допомогою **cin.getline** та збереження його у масив **name**. Використовується **sizeof(name)**, щоб вказати максимальну довжину введеного імені.
4. Визначення довжини введеного імені за допомогою функції **strlen**.
5. Виведення повідомлення "Ім'я в зворотньому порядку:".
6. Запуск циклу **for** для виведення імені у зворотньому порядку. Починаємо з останнього символу індексу **length - 1** та рухаємося назад до першого символу.
7. Виведення кожного символу імені у зворотньому порядку за допомогою **cout**.
8. Завершення програми.
9. #include <iostream>v
10. #include <Windows.h>
11. #include <cstring>
12. using namespace std;
13. int main() {
14. SetConsoleCP(1251);
15. SetConsoleOutputCP(1251);
16. char name[100];
17. cout << "Введіть ваше ім'я: ";
18. cin.getline(name, sizeof(name));
19. int length = strlen(name);
20. cout << "Ім'я в зворотньому порядку: ";
21. for (int i = length - 1; i >= 0; i--) {
22. cout << name[i];
23. }
24. cout << endl;
25. }

Результат:



Задача №3

Умова:

Проєкт3.«Довідник» Скласти навчальну програму, яка перевіряє знання:

4->Назв хімічних елементів( хімія);

Алгоритм:

1. Оголошення змінної **killk** для обліку правильних відповідей та змінної **i** для ітерації.
2. Оголошення масиву вказівників **elements**, який містить символи хімічних елементів, та масиву **names**, який містить назви цих елементів.
3. Оголошення двовимірного масиву **answers** для зберігання введених відповідей користувача.
4. Запуск циклу **for**, який перебирає 3 елементи (елементи, які потрібно вгадати).
5. Виведення повідомлення про елемент із відповідним запитом для введення символу елемента.
6. Зчитування введеного символу в масив **answers[i]** за допомогою **cin.getline**.
7. Порівняння введеного символу з правильним символом елемента (за допомогою **strcmp**). Якщо вони співпадають (повертається 0), то збільшується лічильник **killk**.
8. Виведення заголовку для таблиці результатів.
9. Запуск циклу **for**, щоб вивести правильні символи елементів та введені відповіді користувача.
10. Виведення кількості правильних відповідей.
11. Завершення програми.

Код програми:

#include <iostream>

#include <Windows.h>

#include <cstring>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int killk = 0, i;

const char\* elements[3] = { "H", "He", "Li" };

const char\* names[3] = { "Водень", "Гелій", "Літій" };

char answers[3][20];

for (i = 0; i < 3; i++) {

cout << "\nЕлемент: " << names[i] << "\nВведіть символ елемента: ";

cin.getline(answers[i], sizeof(answers[i]));

if (strcmp(answers[i], elements[i]) == 0)

killk++;

}

cout << "\n\nПравильні відповіді:\tВаші відповіді:\n\n";

for (i = 0; i < 3; i++)

cout << "\t" << elements[i] << "\t\t\t" << answers[i] << endl;

cout << "\n\nКількість правильних відповідей: " << killk << endl;

return 0;

}

Результат:

Задача №4

Умова:

Придумати спосіб шифрування/дешифрування тексту

Алгоритм:

1. Визначається функція з назвою **encryptDecrypt**, яка приймає два параметри: константний посилання на рядок (**text**) і символ (**key**). Ця функція шифрує або дешифрує текст, застосовуючи операцію XOR між кожним символом вхідного тексту та символом ключа. Результат зберігається у новому рядку, який потім повертається.
2. Програма запитує користувача ввести текст, який вони бажають зашифрувати.
3. Користувач також запитується ввести ключ (один символ) для шифрування.
4. Функція **encryptDecrypt** викликається з введеним користувачем текстом і ключем для шифрування, і результат зберігається в змінну **encryptedText**.
5. Зашифрований текст відображається користувачеві.
6. Функція **encryptDecrypt** викликається ще раз, на цей раз з зашифрованим текстом та тим самим ключем для дешифрування.
7. Розшифрований текст відображається користувачеві.

Код програми:

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

string encryptDecrypt(const string& text, const char key) {

string result = text;

for (char& c : result) {

c = c ^ key;

}

return result;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string plaintext;

char encryptionKey;

cout << "Введіть текст для шифрування: ";

getline(cin, plaintext);

cout << "Введіть ключ для шифрування (символ): ";

cin >> encryptionKey;

string encryptedText = encryptDecrypt(plaintext, encryptionKey);

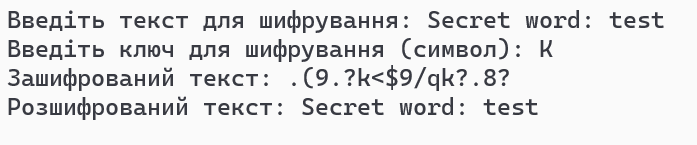
cout << "Зашифрований текст: " << encryptedText << endl;

string decryptedText = encryptDecrypt(encryptedText, encryptionKey);

cout << "Розшифрований текст: " << decryptedText << endl;

}

Результат:



Висновок: Я на цій лабораторній роботі навчився працювати з рядками та масивами символів.