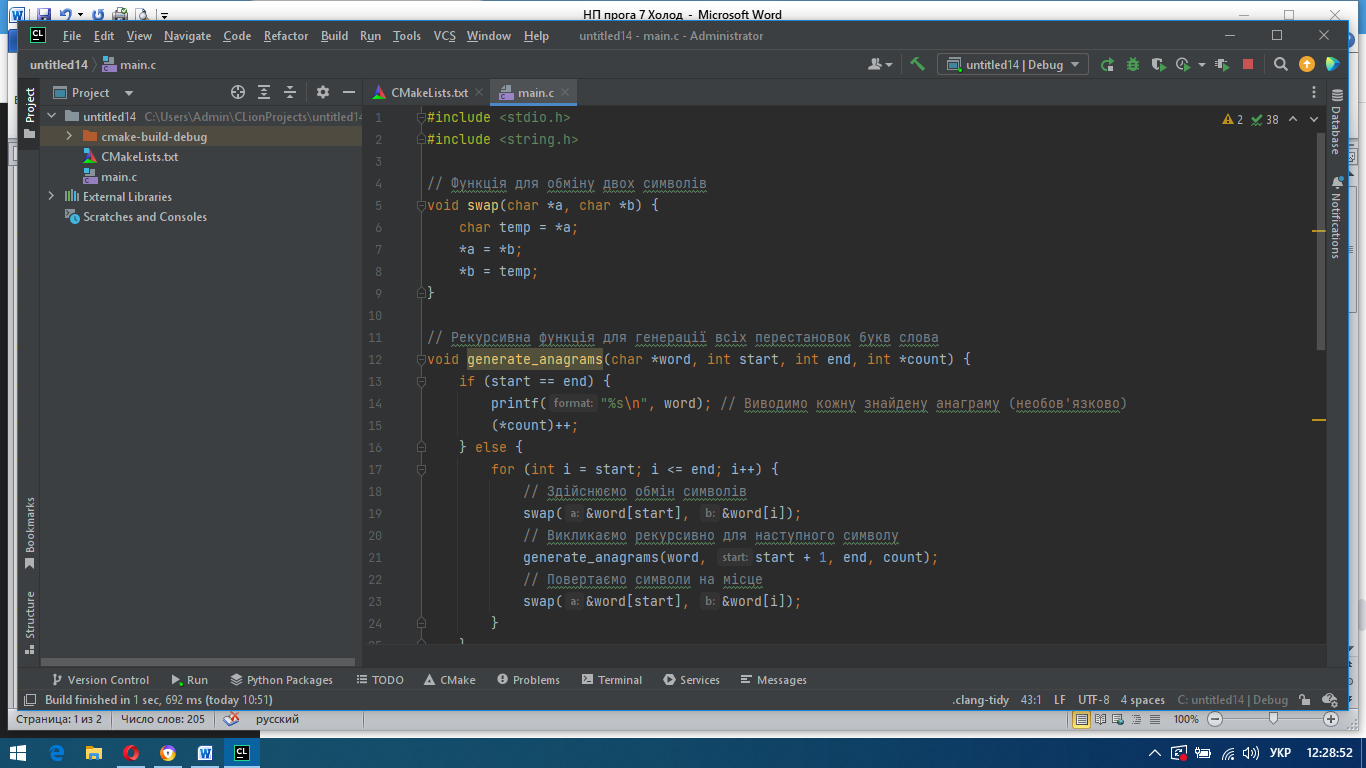
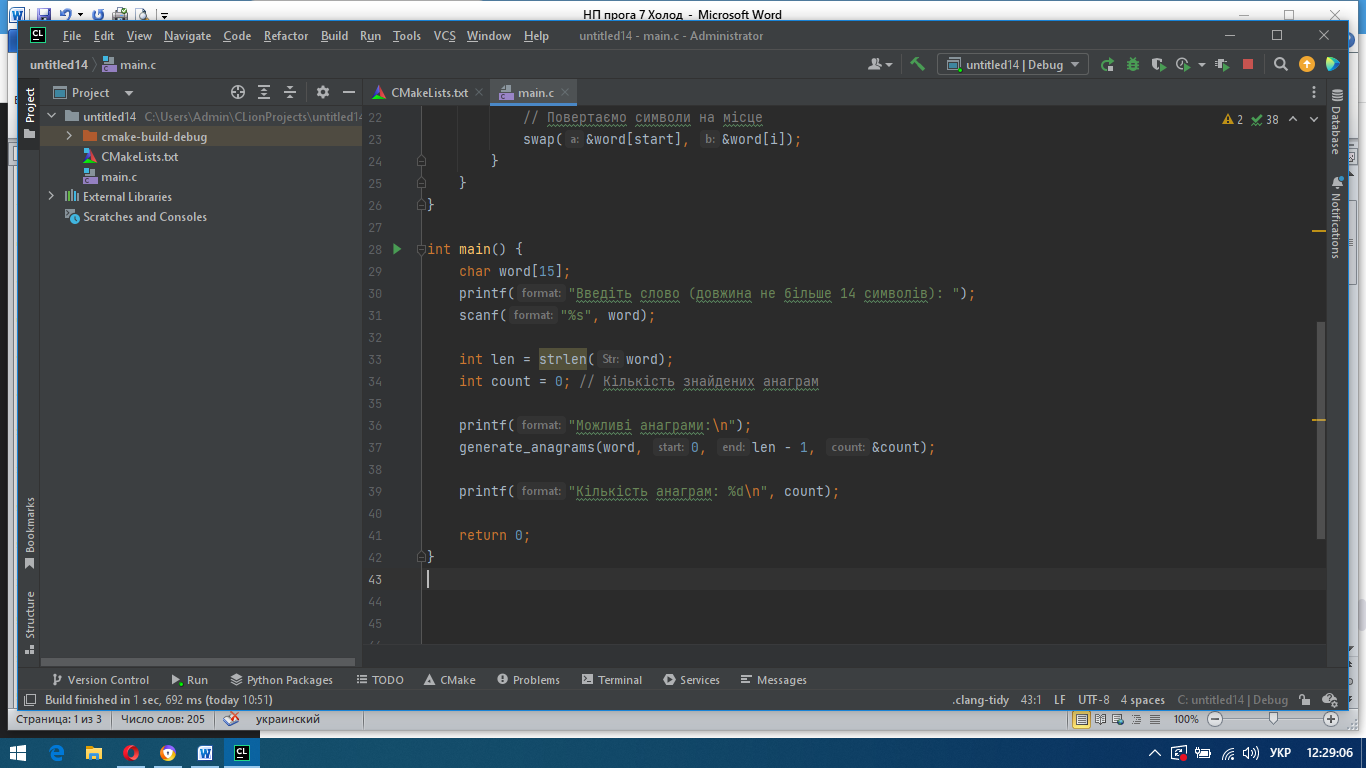
**Завдання:** Анаграмою слова називають перестановку всіх букв слова. Наприклад зі слова SOLO можна отримати 12 анаграм: SOLO, LOSO, OSLO, OLSO, OSOL, OLOS, SLOO, OOLS, OOSL, LOOS, SOOL. напишіть програму, яка виводитиме кількість анаграм , які можуть вийти зі слова.

Вхідні дані: в одному рядку задане слово, кількість букв якого не перевищує 14.

Вихідні дані: кількість можливих анаграм.

**Скрін працюючої програми:**





**Код програми:** #include <stdio.h>

#include <string.h>

// Функція для обміну двох символів

void swap(char \*a, char \*b) {

char temp = \*a;

\*a = \*b;

\*b = temp;

}

// Рекурсивна функція для генерації всіх перестановок букв слова

void generate\_anagrams(char \*word, int start, int end, int \*count) {

if (start == end) {

printf("%s\n", word); // Виводимо кожну знайдену анаграму (необов'язково)

(\*count)++;

} else {

for (int i = start; i <= end; i++) {

// Здійснюємо обмін символів

swap(&word[start], &word[i]);

// Викликаємо рекурсивно для наступного символу

generate\_anagrams(word, start + 1, end, count);

// Повертаємо символи на місце

swap(&word[start], &word[i]);

}

}

}

int main() {

char word[15];

printf("Введіть слово (довжина не більше 14 символів): ");

scanf("%s", word);

int len = strlen(word);

int count = 0; // Кількість знайдених анаграм

printf("Можливі анаграми:\n");

generate\_anagrams(word, 0, len - 1, &count);

printf("Кількість анаграм: %d\n", count);

return 0;

}

**Коментар:** Рекурсивний алгоритм backtracking (повернення назад). Алгоритм буде генерувати всі можливі перестановки букв слова і підраховувати їх кількість.