

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет

ЗВІТ  
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 7  
з навчальної дисципліни  
“Базові методології та технології програмування”  
ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРОБЛЕННЯ МАСИВІВ  
ДАНИХ ТА СИМВОЛЬНОЇ ІНФОРМАЦІЇ

ЗАВДАННЯ ВИДАВ  
доцент кафедри кібербезпеки  
та програмного забезпечення  
Доренський О. П.  
<https://github.com/odorenskyi/>

ВИКОНАВ  
студент академічної групи КН-24  
Булюкін В. Ю.  
<https://github.com/kuroshi-dev>

ПЕРЕВІРИВ  
ст. викладач кафедри кібербезпеки  
та програмного забезпечення  
Коваленко А. С.

## 17 Варіант

**Тема:** Програмна реалізація оброблення масивів даних та символної інформації.

**Мета:** Набуття ґрунтовних вмінь і практичних навичок синтезу алгоритмів оброблення масивів даних та символної (текстової) інформації у кодуваннях UTF-8 і CP866, їх програмної реалізації мовою програмування мовою програмування C (ISO/IEC 9899:2018) задля реалізації програмних засобів у вільному кросплатформовому Code::Blocks IDE.

### Завдання 7.1

Користувач вводить речення (українською або англійською мовою), яке закінчується "/", "/" або "". Вивести повідомлення, чи є у введеному реченні слово "комп'ютер" (без урахування регістру).

```
#include <stdio.h>
#include <string.h>
#include <windows.h>
#include <ctype.h>
#include <locale.h>

#define MAX_LENGTH 1000

void toLowerCase(char *str) {
    for(int i = 0; str[i]; i++) {
        str[i] = tolower(str[i]);
    }
}

int main() {
    SetConsoleCP(65001);
    SetConsoleOutputCP(65001);
    setlocale(LC_ALL, "");

    char sentence[MAX_LENGTH];
    char word[] = "комп'ютер";
    int found = 0;
    int i;

    printf("Введіть речення (завершіть /, // або |): ");
    gets(sentence);

    int len = strlen(sentence);
    if (sentence[len-1] != '/' && sentence[len-1] != '|') {
        if (!(sentence[len-2] == '/' && sentence[len-1] == '/')) {
            printf("Помилка! Речення має закінчуватись /, // або |\n");
            return 1;
        }
    }

    for(i = 0; sentence[i] != '\0'; i++) {
        if(sentence[i] >= 'A' && sentence[i] <= 'Z') {
            sentence[i] = sentence[i] + 32;
        }
    }

    if (strstr(sentence, word) != NULL) {
        found = 1; }
}
```

```

if (found == 1) {
    printf("Слово \"комп'ютер\" знайдено!\n");
} else {
    printf("Слово \"комп'ютер\" не знайдено!\n");
}

printf("Натисніть будь-яку клавішу для завершення...");
getchar();

return 0;
}

```

## Завдання 7.2

Вхід: 10 натуральних чисел

Вихід: кількість чисел, рівних 1, 3, 157, 256 або 1000

```

#include <stdio.h>

int main() {
    int numbers[10];
    int count = 0;

    printf("Введіть 10 натуральних чисел:\n");
    for (int i = 0; i < 10; i++) {
        scanf("%d", &numbers[i]);
    }

    for (int i = 0; i < 10; i++) {
        switch (numbers[i]) {
            case 1:
            case 3:
            case 157:
            case 256:
            case 1000:
                count++;
                break;
        }
    }

    printf("Кількість чисел що дорівнюють 1, 3, 157, 256, або 1000: %d\n", count);

    printf("Натисніть будь-яку клавішу для завершення...");
    getchar();
    return 0;
}

```

## 50 Аргументів

### Чому я навчився або вдосконалив в ході виконання лабораторної роботи

1. Роботі з масивами в мові програмування C.
2. Оголошенню та ініціалізації масивів.
3. Використанню циклів for для роботи з масивами.
4. Перевірці введених даних.
5. Функціям стандартної бібліотеки stdio.h.
6. Функції scanf() для зчитування даних з клавіатури.
7. Функції printf() для виведення даних.

8. Використанню умовних операторів if та else.
9. Оператору switch для перевірки значень.
10. Оператору break для виходу з конструкції switch.
11. Функціям роботи з рядками з бібліотеки string.h.
12. Функції strstr() для пошуку підрядка в рядку.
13. Перетворенню символів у нижній регістр за допомогою функції tolower().
14. Роботі з бібліотекою ctype.h.
15. Використанню функції setlocale() для підтримки української мови.
16. Перевірці символів рядка на верхній регістр.
17. Перетворенню символів з верхнього регістру в нижній.
18. Роботі з масивами символів.
19. Перевірці символів у масиві за допомогою циклу for.
20. Оголошенню та ініціалізації рядків.
21. Використанню макросів #define.
22. Встановленню максимального розміру рядка через макрос.
23. Перевірці закінчення рядка на певні символи.
24. Визначенню довжини рядка через функцію strlen().
25. Роботі з масивами цілих чисел.
26. Підрахунку кількості елементів у масиві.
27. Використанню логічних умов.
28. Ініціалізації змінних.
29. Роботі з вказівниками на рядки.
30. Перевірці масиву символів на наявність підрядка.
31. Формуванню простих алгоритмів для обробки масивів.
32. Оголошенню цілочисельних змінних.
33. Перевірці введених чисел на відповідність умові.
34. Роботі з функцією gets() (хоча її використання не рекомендується).
35. Використанню тернарного оператора.
36. Обробці вихідних даних.
37. Валідації введених даних.
38. Перевірці кількох умов одночасно.
39. Роботі з багатоступеневими умовами.
40. Формуванню повідомлень про помилку.
41. Роботі з бінарними операціями.
42. Форматуванню виводу даних.
43. Поняттю кодування символів.
44. Використанню бібліотеки locale.h.
45. Синтаксису стандарту ISO/IEC 9899:2018.
46. Створенню консольних програм.
47. Розробці алгоритмів пошуку.
48. Використанню циклів для підрахунку.
49. Використанню змінних-лічильників.

50. Підготовці звіту до лабораторної роботи.

### **Висновок**

У ході виконання лабораторної роботи №7 було набуто практичне застосування алгоритмів обробки масивів даних та символьної інформації мовою програмування C (ISO/IEC 9899:2018).

У першому завданні була реалізована програма для пошуку слова "комп'ютер" у введеному користувачем реченні без урахування регістру символів. Було застосовано функції роботи з рядками, зокрема `strstr()`, а також переведення символів у нижній регістр для забезпечення коректного пошуку.

У другому завданні реалізована програма для підрахунку кількості заданих натуральних чисел у масиві за допомогою оператора `switch`. Це дозволило ефективно виконати перевірку на рівність кількох фіксованих значень.

Отримані результати підтвердили правильність роботи програм, а також дозволили закріпити навички роботи з масивами, рядками та умовними операторами.

### **Відповіді на контрольні питання**

1. `git init` — Ініціалізація нового локального репозиторію в поточній папці. Створює приховану папку `.git`, яка містить усі дані для відстеження змін.
2. `git add (git add .)` — Додає файли до індексу (області підготовки) для наступного коміту. `git add .` додає всі файли в поточній директорії та її піддиректоріях.
3. `git commit (git commit -m "текст_коміту")` — Фіксує зміни в репозиторії з описом змін. Опція `-m` дозволяє додати повідомлення про коміт без відкриття текстового редактора.
4. `git remote add (git remote add origin)` — Додає віддалений репозиторій із псевдонімом `origin`, щоб можна було надсилати туди зміни.
5. `git push (git push)` — Надсилає закомічені зміни з локального репозиторію на віддалений репозиторій, прив'язаний до гілки.