

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”  
ІНСТИТУТ КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ, АВТОМАТИКИ ТА  
МЕТРОЛОГІЇ  
КАФЕДРА КОМП’ЮТЕРИЗОВАНИХ СИСТЕМ АВТОМАТИКИ



**ЗВІТ**

про виконання лабораторної роботи № 2

з навчальної дисципліни:

**«Алгоритмізація та програмування, частина 3»**

*Виконав:*

*студент групи ІР-24*

*Шийка Андрій*

*Прийняв:*

*Клиско Ю.В.*

*Львів - 2025*

**Мета роботи:** ознайомитися з формами визначення одновимірних масивів у мові Cі, використанням покажчиків при роботі з масивами, навчитися формувати числові і символьні масиви.

#### Завдання 4.1.

Розробіть програму, в якій з клавіатури вводиться масив цілих чисел заданого розміру, виконуються задані обчислення. На екран виводяться значення елементів масиву і результати обчислень.

Обчисліть суму квадратів від'ємних елементів. Розмір масиву 7.

```
#include <stdio.h>

#include <stdlib.h>

#define SIZE 7

int main()
{
    int arr[SIZE];
    int sum = 0;
    int i;

    // Введення елементів масиву з клавіатури
    printf("Введіть %d цілих чисел:\n", SIZE);
    for (i = 0; i < SIZE; i++)
    {
        printf("arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    // Виведення елементів масиву
    printf("\nЗначення елементів масиву:\n");
    for (i = 0; i < SIZE; i++)
```

```
{

    printf("arr[%d] = %d\n", i, arr[i]);

}

// Обчислення суми квадратів від'ємних елементів

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

{

    if (arr[i] < 0)

    {

        sum += arr[i] * arr[i];

    }

}

// Виведення результату

printf("\nСума квадратів від'ємних елементів: %d\n", sum);

return 0;

}
```

```

● > ./lab_2/4_1
Введіть 7 цілих чисел:
arr[0] = -1
arr[1] = -2
arr[2] = 2
arr[3] = 3
arr[4] = 5
arr[5] = 6
arr[6] = 9

Значення елементів масиву:
arr[0] = -1
arr[1] = -2
arr[2] = 2
arr[3] = 3
arr[4] = 5
arr[5] = 6
arr[6] = 9

Сума квадратів від'ємних елементів: 5
ap_p3_labs_iot_nulp on ʝ lab/2 took 15s
○ >

```

## Завдання 4.2.

Розробіть програму, яка виконує задані обчислення та виводить на екран масив і результат обчислення.

Заповніть масив цілих чисел випадковими числами від -30 до 30 та обчисліть кількість непарних чисел. Розмір масиву 14.

```

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIZE 14

int main()
{
    int arr[SIZE];

    int i, count = 0;

```

```
// Запуск генератора випадкових чисел

srand((unsigned)time(NULL));

// Заповнення масиву випадковими числами від -30 до 30

printf("Елементи масиву:\n");

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

{

    arr[i] = rand() % 61 - 30; // діапазон від -30 до +30

    printf("arr[%d] = %3d\n", i, arr[i]);

}

// Підрахунок кількості непарних чисел

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

{

    if (arr[i] % 2 != 0)

    { // перевірка на непарність

        count++;

    }

}

// Виведення результату

printf("\nКількість непарних чисел: %d\n", count);

return 0;

}
```

```
● > ./lab_2/4_2
Елементи масиву:
arr[0] = 28
arr[1] = -17
arr[2] = 26
arr[3] = -13
arr[4] = -12
arr[5] = 28
arr[6] = -20
arr[7] = -14
arr[8] = 14
arr[9] = 29
arr[10] = 8
arr[11] = -9
arr[12] = 10
arr[13] = -28

Кількість непарних чисел: 4
ap_p3_labs_iot_nulp on lab/2
○ > 
```

### Завдання 4.3.

Розробіть програму, в якій визначається масив цілих чисел заданого розміру. Перший елемент масиву задається, всі інші елементи вводяться з клавіатури, потім виконуються задані обчислення, використовуючи метод доступу до елементів масиву через розіменування покажчика. На екран виводяться значення елементів масиву і результати обчислень.

Визначте мінімальний непарний елемент масиву. Перший елемент масиву дорівнює 201. Розмір масиву 7.

```
#include <stdio.h>

#include <stdlib.h>

#define SIZE 7

int main()
{
    int arr[SIZE];

    int *px = arr; // покажчик на масив

    int i;

    int min_odd;

    int found = 0; // прапорець для перевірки наявності непарних елементів
```

```
// Перший елемент масиву дорівнює 201

*px = 201;

// Введення решти елементів масиву з клавіатури
printf("Перший елемент arr[0] = %d (заданий)\n", *px);
printf("\nВведіть ще %d елементів масиву:\n", SIZE - 1);

for (i = 1; i < SIZE; i++)
{
    printf("arr[%d] = ", i);

    scanf("%d", px + i); // доступ через покажчик
}

// Виведення всіх елементів масиву
printf("\nЗначення елементів масиву:\n");
for (i = 0; i < SIZE; i++)
{
    printf("arr[%d] = %d\n", i, *(px + i));
}

// Пошук мінімального непарного елемента
for (i = 0; i < SIZE; i++)
{
    if (*(px + i) % 2 != 0)
    { // перевірка на непарність
        if (!found)
        {
            min_odd = *(px + i); // перший непарний елемент
            found = 1;
        }
    }
}
```

```

    }

    else if (*(px + i) < min_odd)

    {

        min_odd = *(px + i); // оновлення мінімуму

    }

}

}

// Виведення результату

if (found)

{

    printf("\nМінімальний непарний елемент: %d\n", min_odd);

}

else

{

    printf("\nУ масиві немає непарних елементів.\n");

}

return 0;

}

```

```

● > ./lab_2/4_3
Перший елемент arr[0] = 201 (заданий)

```

Введіть ще 6 елементів масиву:

```

arr[1] = 2
arr[2] = 3
arr[3] = 203
arr[4] = 35
arr[5] = 291
arr[6] = 891

```

Значення елементів масиву:

```

arr[0] = 201
arr[1] = 2
arr[2] = 3
arr[3] = 203
arr[4] = 35
arr[5] = 291
arr[6] = 891

```

Мінімальний непарний елемент: 3



#### Завдання 4.4.

Розробіть програму, що сортує заданим методом одновимірний масив цілих чисел та виводить масив до і після сортування.

Заповніть масив цілих чисел випадковими числами від -50 до 50 та відсортуйте за збуванням масив чисел методом прямого упорядкування.

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 10

int main()

{

    int dig[N];

    int i, j, mm, max, tmp;

    // Запуск генератора випадкових чисел

    srand((unsigned)time(NULL));

    // Заповнення масиву випадковими числами від -50 до 50

    printf("Заповнення масиву випадковими числами від -50 до 50\n\n");

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

    {

        dig[i] = rand() % 101 - 50; // діапазон від -50 до +50

    }

    // Виведення масиву до сортування

    printf("Масив до сортування:\n");

    for (i = 0; i < N; i++)
```

```
{

    printf("%4d ", dig[i]);

}

printf("\n\n");

// Сортуння методом прямого упорядкування (метод вибору) за УБУВАННЯМ
for (i = 0; i < N - 1; i++)
{

    max = dig[i]; // припускаємо, що перший елемент максимальний
    mm = i;       // запам'ятовуємо його індекс

    // Шукаємо максимальний елемент у невідсортованій частині
    for (j = i + 1; j < N; j++)
    {

        if (dig[j] > max)

        {

            max = dig[j];

            mm = j;

        }

    }

    // Обмін місцями максимального елемента з поточним
    if (mm != i)
    {

        tmp = dig[i];

        dig[i] = dig[mm];

        dig[mm] = tmp;

    }

}
```

```
// Виведення масиву після сортування

printf("Масив ПІСЛЯ сортування за убунанням:\n");

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

{

    printf("%4d ", dig[i]);

}

printf("\n");

return 0;

}
```

```
● > ./lab_2/4_4
Заповнення масиву випадковими числами від -50 до 50

Масив ДО сортування:
-21 -25 -37 -29 -17 50 -33 48 -26 44

Масив ПІСЛЯ сортування за убунанням:
50 48 44 -17 -21 -25 -26 -29 -33 -37

ap_p3_labs_iot_nulp on ʘ lab/2
● > ./lab_2/4_4
Заповнення масиву випадковими числами від -50 до 50

Масив ДО сортування:
-40 4 -8 20 -12 47 10 4 -4 15

Масив ПІСЛЯ сортування за убунанням:
47 20 15 10 4 4 -4 -8 -12 -40
```

## Завдання 4.5.

Розробіть програму, в якій вводиться рядок з кількома словами, підраховується кількість символів в ньому, визначається рядок з заданим повідомленням і теж підраховується кількість символів в ньому. Потім до першого рядка додається повідомлення, і на екран виводиться сумарний рядок та кількість символів в ньому.

Введіть рядок з назвою книги і прізвищем автора та повідомте, що така книга з'явилася в бібліотеці.

```
#include <stdio.h>

#include <string.h>

#define K 80
```

```
int main()
{
    char str1[K], str2[K], result[K * 2]; // резервування пам'яті для рядків
    int len1, len2, lenResult;

    // Введення першого рядка

    puts("Введіть рядок з кількома словами:");

    fgets(str1, K, stdin); // безпечніше, ніж gets()

    // Видаляємо символ нового рядка, якщо він є
    str1[strcspn(str1, "\n")] = '\0';

    // Підрахунок символів у першому рядку
    len1 = strlen(str1);

    printf("\nПерший рядок: %s\n", str1);
    printf("Кількість символів у першому рядку: %d\n", len1);

    // Визначення другого рядка з заданим повідомленням
    strcpy(str2, " з'явилася в бібліотеці");

    // Підрахунок символів у другому рядку
    len2 = strlen(str2);

    printf("\nДругий рядок (повідомлення): %s\n", str2);
    printf("Кількість символів у другому рядку: %d\n", len2);

    // Об'єднання рядків
    strcpy(result, str1);
    strcat(result, str2);
}
```

```

// Підрахунок символів у результуючому рядку

lenResult = strlen(result);

// Виведення результату

printf("\n=== РЕЗУЛЬТАТ ===\n");

printf("Сумарний рядок: %s\n", result);

printf("Кількість символів у сумарному рядку: %d\n", lenResult);

return 0;
}

```

● > ./lab\_2/4\_5

Введіть рядок з кількома словами:

Танець\_недоумка Ілларіон\_Павлюк

Перший рядок: Танець\_недоумка Ілларіон\_Павлюк

Кількість символів у першому рядку: 59

Другий рядок (повідомлення): з'явилася в бібліотеці

Кількість символів у другому рядку: 42

=== РЕЗУЛЬТАТ ===

Сумарний рядок: Танець\_недоумка Ілларіон\_Павлюк з'явилася в бібліотеці

Кількість символів у сумарному рядку: 101

ap\_p3\_labs\_iot\_nulp on ʘ lab/2 took 1m13s

○ > █