

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота №6
з дисципліни «Алгоритмізація та програмування»
на тему «Реалізація алгоритмів обробки одновимірних масивів
мовою C ++»

XAI.301.175.318.13 ЛР

Виконав _____ студент гр. № 319а
_____ Михайло ТЮТЮННИК
(підпис, дата) (П.І.Б.)

Перевірив к.т.н., доцент
(вчена ступінь, вчене звання)
_____ Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення одновимірних і масивів на мові C ++ і реалізувати декларацію, введення з консолі, обробку і виведення в консоль одновимірних масивів на мові C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1: Вивести всі парні числа масиву в порядку зменшення їх індексів та визначити їх кількість.

Завдання 2: Реверсувати масив (змінити порядок елементів на зворотний).

Завдання 3: Організувати багаторазовий вибір одного з двох завдань у головній функції

ВИКОНАННЯ РОБОТИ

Завдання 1: Виведення парних чисел у порядку зменшення індексів

Вхідні дані:

Масив цілих чисел, введений користувачем.

Розмір масиву: ціле число N ($2 \leq N \leq 50$).

Вихідні дані:

Парні числа масиву, виведені в порядку зменшення їх індексів.

Кількість парних чисел (K).

Алгоритм:

1. Введення розміру масиву (N).
2. Введення елементів масиву.
3. Пошук парних чисел у масиві та їх виведення у зворотному порядку.
4. Підрахунок кількості парних чисел (K).

Лістинг коду до завдання 1 наведено в дод. А (стор. 4)

Зображення вікна виконання роботи наведено в дод. Б(стор.6)

Завдання 2: Реверсування масиву

Вхідні дані:

Масив цілих чисел, введений користувачем.

Розмір масиву: ціле число N ($2 \leq N \leq 50$).

Вихідні дані:

Масив з елементами, розташованими у зворотному порядку.

Алгоритм:

1. Введення розміру масиву (N).
2. Введення елементів масиву.
3. Зміна порядку елементів масиву на зворотний.
4. Виведення реверсованого масиву.

Лістинг коду до завдання 1 наведено в дод. А (стор. 4)

Зображення вікна виконання роботи наведено в дод. Б(стор.6)

Завдання 3: Багаторазовий вибір завдань

Вхідні дані:

Масив цілих чисел, введений користувачем.

Розмір масиву: ціле число N ($2 \leq N \leq 50$).

Вихідні дані:

Результати виконання обраного завдання (завдання 1 або завдання 2).

Алгоритм:

1. Введення розміру масиву (N).
2. Введення елементів масиву.
3. Виведення меню для вибору завдання.
4. Виконання обраного завдання.

Лістинг коду до завдання 1 наведено в дод. А (стор. 4)

Зображення вікна виконання роботи наведено в дод. Б(стор.6)

ВИСНОВКИ

У ході виконання лабораторної роботи було вивчено основи роботи з одновимірними масивами на мові C++. Було реалізовано програму, яка дозволяє виконувати наступні операції:

1. Виведення парних чисел масиву у порядку зменшення їх індексів.
2. Реверсування масиву.
3. Організація багаторазового вибору завдань у головній функції.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
using namespace std;
// Функція для введення масиву
void inputArray(int* arr, int N) {
    cout << "Enter the elements of the array: \n";
    for (int i = 0; i < N; ++i) {
        cin >> arr[i];
    }
}
// Функція для виведення масиву
void outputArray(int* arr, int N) {
    for (int i = 0; i < N; ++i) {
        cout << arr[i] << " ";
    }
    cout << endl;
}
// Завдання 1: Вивести числа, кратні 10, у порядку зменшення індексів
void task1(int* arr, int N) {
    int* multiples_of_10 = new int[N];
    int count = 0;
    for (int i = N - 1; i >= 0; --i) {
        if (arr[i] % 10 == 0) {
            multiples_of_10[count++] = arr[i];
        }
    }
    cout << "Numbers divisible by 10 (in reverse index order): ";
    for (int i = 0; i < count; ++i) {
        cout << multiples_of_10[i] << " ";
    }
    cout << endl;
    cout << "Count of numbers divisible by 10 (K): " << count << endl;
    delete[] multiples_of_10;
}
// Завдання 2: Поміняти порядок елементів на зворотний
void task2(int* arr, int N) {
    for (int i = 0; i < N / 2; ++i) {
        int temp = arr[i];
        arr[i] = arr[N - 1 - i];
        arr[N - 1 - i] = temp;
    }
    cout << "Reversed array: ";
    outputArray(arr, N);
}
int main() {
    int N;
    cout << "Enter the size of the array (N): ";
    cin >> N;
    int* arr = new int[N];
    inputArray(arr, N);
    int choice;
    do {
        cout << "\nChoose a task:\n";
        cout << "1. Find numbers divisible by 10\n";
        cout << "2. Reverse the array\n";
        cout << "0. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
            case 1:
                task1(arr, N);
                break;
            case 2:

```

```
        task2(arr, N);
        break;
    case 0:
        cout << "Exiting program." << endl;
        break;
    default:
        cout << "Invalid choice. Please try again." << endl;
    }
} while (choice != 0);
delete[] arr;
return 0;
}
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

На рисунку 1 зображена робота програми

```

Enter the size of the array (N): 5
Enter the elements of the array: 90 87 50 21 10
Choose a task:
1. Find numbers divisible by 10
2. Reverse the array
0. Exit
Enter your choice: 1
Numbers divisible by 10 (in reverse index order): 10 50 90
Count of numbers divisible by 10 (K): 3 the elements of the array: \n";
Choose a task:
1. Find numbers divisible by 10
2. Reverse the array
0. Exit
Enter your choice: 2
Reversed array: 10 21 50 87 90
Choose a task:
1. Find numbers divisible by 10
2. Reverse the array
0. Exit
Enter your choice: 0
Exiting program.
C:\Users\User\source\repos\Project3\x64\Debug\Project3.exe (процесс 13624) завершил работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:== 0) {

```

Рисунок 1 — Результат роботи програми

ДОДАТОК В

Діаграми активності програм

На рисунку 1 зображена діаграма активності всієї програми

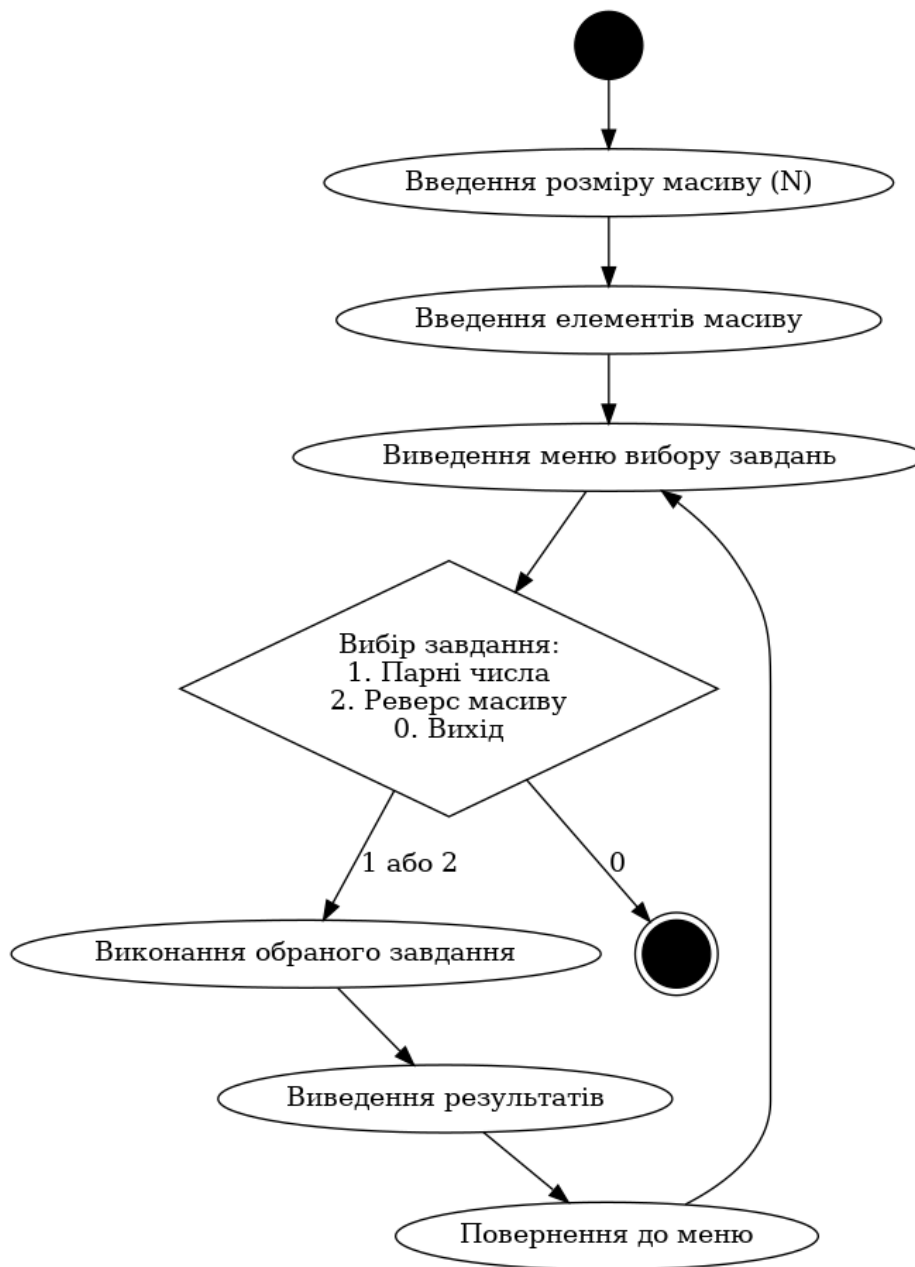


Рисунок 1 – Діаграма активності програми

На рисунку 2 зображена діаграма активності для функції аналізу масиву

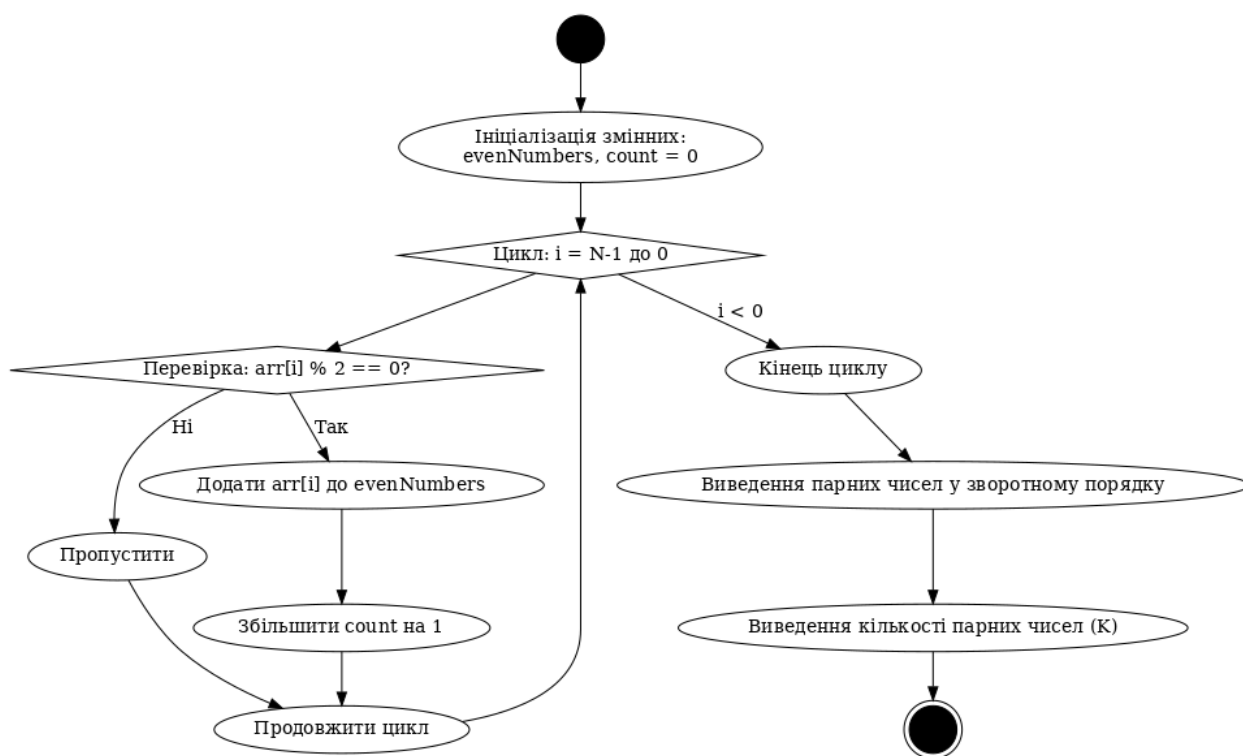


Рисунок 2 Діаграма активності для функції аналізу масиву