# 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 С ++»

# ХАІ.301. 175. 318. ЛР

Виконав студент гр. 318

Нікіта КАРТАВИХ

(підпис, дата) (П.І.Б.)

Перевірив

к.т.н., доц. Олена ГАВРИЛЕНКО

(підпис, дата) (П.І.Б.)

2025

# МЕТА РОБОТИ

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

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів одновимірного масиву. Розмір масиву і його елементів ввести з консолі. Спершу вивести весь масив у рядок в порядку зростання індексів, потім – елементи чи підраховані результати відповідно до завдання.

Array34. Дан масив розміру N. Знайти максимальний з його локальних мінімумів

Завдання 2. Вирішити завдання на перетворення одновимірного масиву. Розмір масиву і його елементи ввести з консолі. Спершу вивести у консоль заданий масив, потім – змінений.

Array85. Дано масив A розміру N і ціле число K (1 ≤ K ≤ 4, K < N). Здійснити циклічний зсув елементів масиву вправо на K позицій (при цьому A1 перейде в AK + 1, A2 - в AK + 2, ..., AN - в AK). Допускається використовувати допоміжний масив з 4 елементів.

Завдання 3. У функції організувати багаторазовий вибір одного з двох завдань. Кожне завдання описати окремою функцією без параметрів. Введення, виведення, обробку масивів реалізувати окремими функціями з параметрами.

# ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Array 34.

Вхідні дані (ім’я, опис, тип, обмеження): N – кількість елементів у масиві, int, N>0.

A – елементи масиву, int, i = 0..N-1.

Вихідні дані (ім’я, опис, тип):

maxLocalMin – Найбільший з локальних мінімумів, якщо такий існує, int або string (повідомлення).

Лістинг коду вирішення задачі Array 34 наведено в дод. А (стор. 5-7). Екран роботи програми показаний на рис. Б.1. (дод. Б, стор. 8-10)

Приклад діаграми для завдання Array 34 наведено на рис. Б.3. (дод. Б, стор.

8-10)

Завдання 2.

Вирішення задачі Array 85.

Вхідні дані (ім’я, опис, тип, обмеження):

N – кількість елементів у масиві, int, N>0.

A – елементи масиву, int, i = 0..N-1.

K – кількість позицій для зсуву, int, 1 ≤ K ≤ 4, K < N.

Вихідні дані (ім’я, опис, тип):

A[i] – Масив після циклічного зсуву вправо на K позицій, int[].

Лістинг коду вирішення задачі Array 85 наведено в дод. А (стор. 5-7). Екран роботи програми показаний на рис. Б.2. (дод. Б, стор. 8-10)

Приклад діаграми для завдання Array 85 наведено на рис. Б.4. (дод. Б, стор.

8-10)

Завдання 3. Організація меню.

Вхідні дані (ім’я, опис, тип, обмеження):

choice – Номер вибраного пункту меню, int,

0 ≤choice ≤ 2.

Вихідні дані (ім’я, опис, тип):

message – повідомлення про вибір задачі або завершення програми, string.

result – результат виконання вибраної задачі або її виклик.

Лістинг коду вирішення завдання 3 наведено в дод. А (стор. 5-7)

Приклад діаграми для завдання 3 наведено на рис. Б.5. (дод. Б, стор. 8-10)

# ВИСНОВКИ

Було вивчено теоретичний матеріал з основ представлення одновимірних і масивів на мові С ++ і було закріплено на практиці реалізацію декларації, введення з консолі, обробку і виведення в консоль одновимірних масивів на мові C ++ в середовищі Visual Studio.

# ДОДАТОК А

Лістинг коду програми

#include <iostream>

#include <vector>

#include <limits>

using namespace std;

// Функція отримання масиву від користувача

vector<int> arrayFromUser(int& n) {

Step1:

cout << "Введіть кількість елементів: ";

cin >> n;

if (cin.fail()) { // Перевірка на коректність введеного значення

cout << "\nКількість елементів повинна бути числом\n";

cin.clear();

cin.ignore();

goto Step1; // Повернення до введення

}

if (n <= 0) { // Перевірка на те, що n > 0

cout << "\nКількість елементів має бути більше нуля\n";

cin.clear();

cin.ignore();

goto Step1; // Повернення до введення

}

cout << "Введіть елементи масиву: ";

vector<int> arrayFromUser(n); // Ініціалізація вектора з n елементами

for (int i = 0; i < n; i++) {

cin >> arrayFromUser[i]; // Введення елементів масиву

}

return arrayFromUser; // Повернення вектора

}

// Функція для виведення масиву на екран

void printArrayToConsole(vector<int> array) {

int n = array.size(); // Отримання розміру масиву

for (int i = 0; i < n; i++) {

cout << array[i] << " "; // Виведення кожного елемента

}

cout << "\n";

}

// Функція отримання значення k від користувача

int kFromUser() {

Step1:

cout << "Введіть на скільки елементів потрібно зсунути масив: ";

int k;

cin >> k;

if (cin.fail()) { // Перевірка на коректність введеного значення

cout << "\n\"k\" повинно бути числом\n";

cin.clear();

cin.ignore();

goto Step1; // Повернення до введення

}

if (k <= 0) { // Перевірка на те, що k > 0

cout << "\n\"k\" має бути більше нуля\n";

cin.clear();

cin.ignore();

goto Step1; // Повернення до введення

}

return k; // Повернення значення k

}

// Функція пошуку найбільшого локального мінімуму

int findMaxLocalMinimum(vector<int>& arrWithNumbers, int& n) {

int maxLocalMin = -numeric\_limits<int>::infinity(); // Початкове значення максимуму

for (int i = 0; i < n; ++i) {

// Перевірка для першого елемента

if (i == 0 && arrWithNumbers[i] < arrWithNumbers[i + 1] && arrWithNumbers[i] > maxLocalMin) {

maxLocalMin = arrWithNumbers[i];

continue;

}

// Перевірка для останнього елемента

if (i == n - 1 && arrWithNumbers[i] < arrWithNumbers[i - 1] && arrWithNumbers[i] > maxLocalMin) {

maxLocalMin = arrWithNumbers[i];

continue;

}

// Перевірка для внутрішніх елементів

if ((arrWithNumbers[i] < arrWithNumbers[i - 1] || arrWithNumbers[i] < arrWithNumbers[i + 1])

&& arrWithNumbers[i] > maxLocalMin) {

maxLocalMin = arrWithNumbers[i];

}

}

return maxLocalMin; // Повернення найбільшого локального мінімуму

}

// Функція для виконання пошуку локального мінімуму

void maxLocalMinimum() {

int n, maxLocalMin;

vector<int> array = arrayFromUser(n); // Отримання масиву від користувача

maxLocalMin = findMaxLocalMinimum(array, n); // Пошук найбільшого локального мінімуму

if (maxLocalMin == -numeric\_limits<int>::infinity())

cout << "Локальних мінімумів не існує\n";

else

cout << "Найбільший локальний мінімум дорівнює: " << maxLocalMin << "\n";

}

// Функція для циклічного зсуву масиву вправо

void cyclicShiftRight() {

int n, k;

vector<int> arrayWithNumbers = arrayFromUser(n); // Отримання масиву від користувача

k = kFromUser(); // Отримання значення k від користувача

vector<int> tempArray(k); // Допоміжний масив для збереження останніх k елементів

// Збереження останніх k елементів

for (int i = 0; i < k; i++) {

tempArray[i] = arrayWithNumbers[n - k + i];

}

// Зсув решти елементів вправо

for (int i = n - 1; i >= k; i--) {

arrayWithNumbers[i] = arrayWithNumbers[i - k];

}

// Вставка k елементів на початок

for (int i = 0; i < k; i++) {

arrayWithNumbers[i] = tempArray[i];

}

printArrayToConsole(arrayWithNumbers); // Виведення результату

}

// Головна функція програми

int main() {

Step1:

int choice;

// Виведення меню вибору

cout << "\nОберіть задачу для виконання:\n";

cout << "1. Пошук найбільшого локального мінімуму\n";

cout << "2. Зсув на k елементів вправо\n";

cout << "0. Вийти\n";

cout << "Ваш вибір: ";

cin >> choice;

// Обробка вибору користувача

switch (choice) {

case 1:

maxLocalMinimum(); // Виклик функції для пошуку локального мінімуму

goto Step1; // Повернення до меню

break;

case 2:

cyclicShiftRight(); // Виклик функції для зсуву масиву

goto Step1; // Повернення до меню

break;

case 0:

cout << "Вихід з програми.\n";

break;

default:

cout << "Некоректний вибір. Спробуйте ще раз.\n";

goto Step1; // Повернення до меню

}

return 0; // Завершення програми

}

# ДОДАТОК Б

Скрін-шоти вікна виконання програми

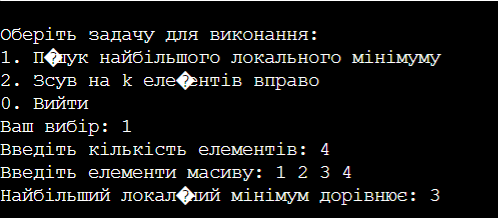


Рисунок Б.1 – Екран виконання програми для вирішення завдання Array 34

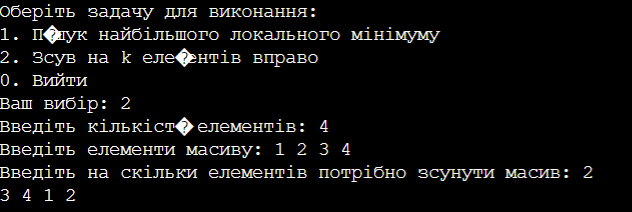


Рисунок Б.2 – Екран виконання програми для вирішення завдання Array 85

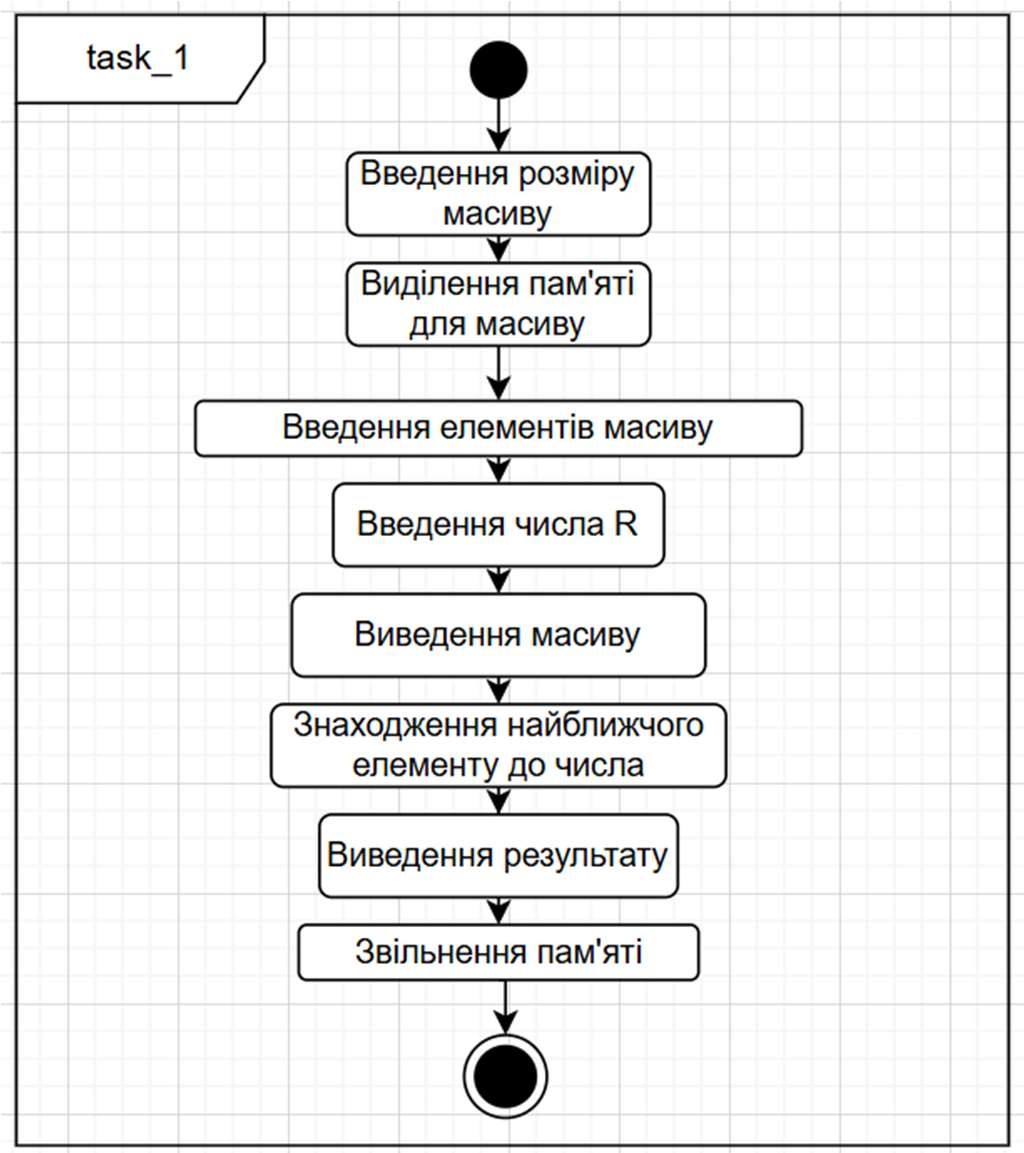


Рисунок Б.3 – Діаграма для завдання Array 40

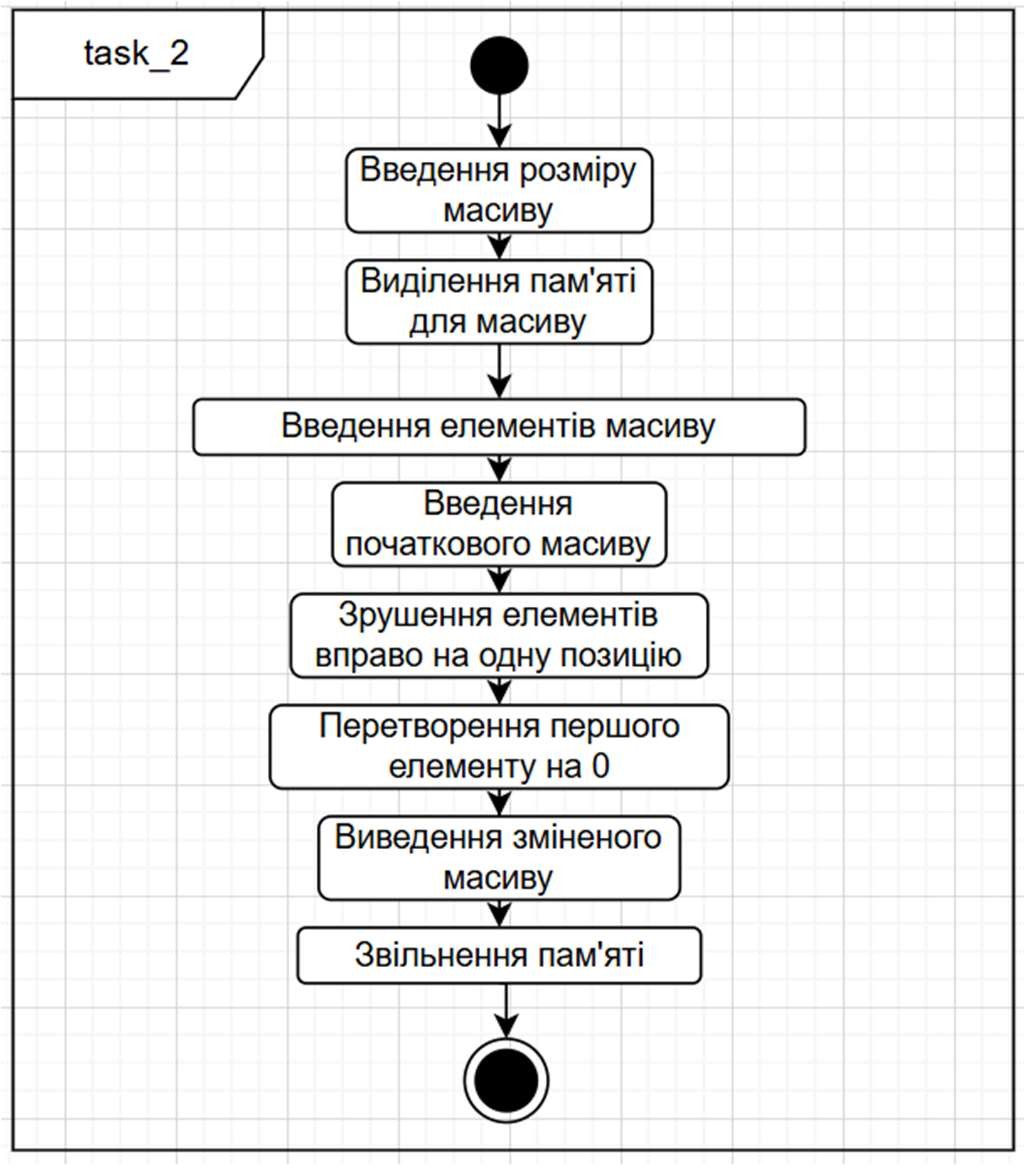


Рисунок Б.4 – Діаграма для завдання Array 79

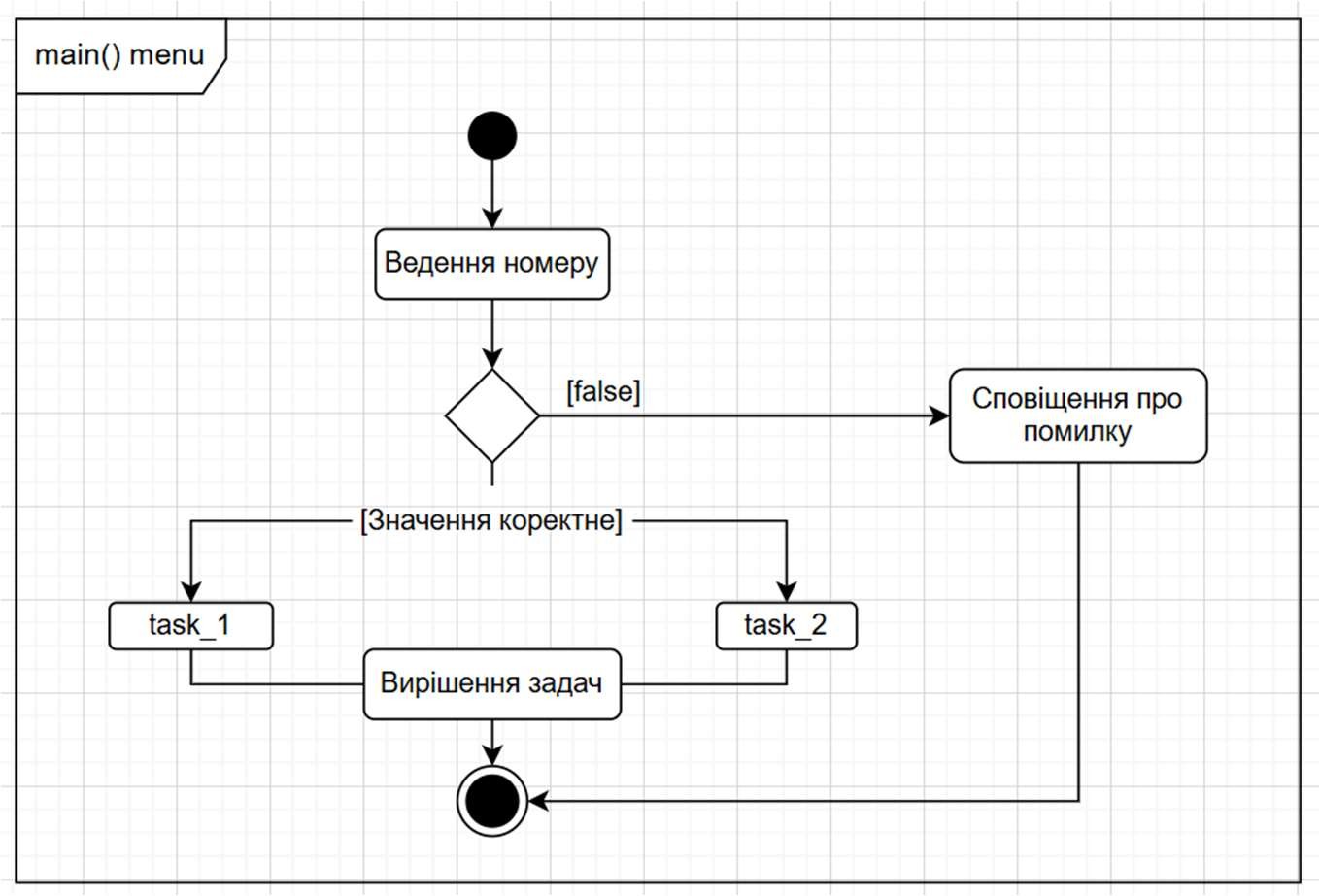


Рисунок Б.5 – Діаграма для завдання 3