

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. 08 ЛР

Виконав студент гр. _____ 318

_____ Каріна ГЛІБОВА
(підпис, дата) (П.І.Б.)

Перевірів

_____ к.т.н., доц. 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. 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.

Array 40. Дано число R і масив A розміру N. Знайти елемент масиву, який найбільш близький до числа R (тобто такий елемент AK, для якого величина $|AK - R|$ є мінімальною).

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

Array 79. Дан масив розміру N. Здійснити зрушення елементів масиву вправо на одну позицію (при цьому A1 перейде в A2, A2 - в A3, ..., AN-1 - в AN, а початкове значення останнього елемента буде втрачено). Перший елемент отриманого масиву покласти рівним 0.

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 40.

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

N – розмір масиву, дійсне число.

A – елементи масиву, дійсні числа.

R – введене число, дійсний тип.

Вихідні дані (ім'я, опис, тип):

closestElement – найближчий елемент.

Лістинг коду вирішення задачі Array 40 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 40 наведено на рис. Б.3. (дод. Б, стор. 8-10)

Завдання 2.

Вирішення задачі Array 79.

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

N – розмір масиву, дійсне число.

Вихідні дані (ім'я, опис, тип):

A[i] – змінений масив.

Лістинг коду вирішення задачі Array 79 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 79 наведено на рис. Б.4. (дод. Б, стор. 8-10)

Завдання 3.

Організація меню.

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

«Номер завдання» – введення номеру завдання.

Вихідні дані (ім'я, опис, тип):

task_1 – якщо ввели число «1», виводяться розрахунки задачі Array 40.

task_2 – якщо ввели число «2», виводяться розрахунки задачі Array 79.

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 C ++ і було закріплено на практиці реалізація декларації, введення з консолі, обробка і виведення в консоль одновимірних масивів на мові C ++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```

#include <iostream>
#include "windows.h"
#include <cmath>
using namespace std;

//Array40. Дано число R і масив A розміру N. Знайти елемент масиву, який
найбільш
//близький до числа R (тобто такий елемент AK, для якого величина | AK - R | є
мінімальною).
void task_1(); //завдання 1, декларація функції

//Array79. Дан масив розміру N. Здійснити зрушення елементів масиву вправо на
одну позицію
//(при цьому A1 перейде в A2, A2 - в A3, ..., AN-1 - в AN, а початкове значення
останнього
//елемента буде втрачено). Перший елемент отриманого масиву покласти рівним 0.
void task_2(); //завдання 2, декларація функції

int main() {
    SetConsoleOutputCP(1251);
    int menu; // зміна для номеру завдання
    do
    { // початок циклу
        cout << "Номер завдання:"; //введення номеру завдання
        cin >> menu; // обирає номеру завдання
        cout << endl; //вільна строка
        switch (menu) {
            case 1: task_1(); break; // 1 - завдання 1
            case 2: task_2(); break; // 2 - завдання 2
            case -1: cout << "Вихід..." << endl; break; // -1 - вихід
            default: cout << "Помилка! Лише 1, 2!" << endl; // інший номер -
повторити
        }
        cout << endl; // вільна строка
        cout << "+-----+" << endl; // строка задля полегшення
візуального сприймання тексту
        cout << endl; // вільна строка
    } // кінець циклу
    while (menu != -1); // умова виконання циклу
    return 0;
}

void task_1() {
    int N, R;

```

```

cout << "Введіть розмір масиву N: "; // введення розміру масиву
cin >> N;

int* A = new int[N]; // динамічне виділення пам'яті для масиву

cout << "Введіть елементи масиву: ";
for (int i = 0; i < N; i++) {
    cin >> A[i]; // введення елементів масиву
}

cout << "Введіть число R: "; // введення числа
cin >> R;

cout << "Масив: "; // виведення масиву
for (int i = 0; i < N; i++) {
    cout << A[i] << " ";
}
cout << endl;

// знаходження найближчого елементу до числа R
int closestElement = A[0]; // початково найближчий елемент
int minDiff = abs(A[0] - R); // початкова різниця (відстань) між першим
елементом та R

for (int i = 1; i < N; i++) {
    int currentDiff = abs(A[i] - R); // поточна різниця
    if (currentDiff < minDiff) {
        minDiff = currentDiff; // оновлена мінімальна різниця
        closestElement = A[i]; // оновлений найближчий елемент
    }
}

// виведення результату
cout << "Елемент масиву, найбільш близький до числа R: " << closestElement
<< endl;

delete[] A; // звільняємо пам'ять після використання масиву
}

void task_2() {
    int N;

    cout << "Введіть розмір масиву N: "; // введення розміру масиву
    cin >> N;

    int* A = new int[N]; // динамічне виділення пам'яті для масиву

    cout << "Введіть елементи масиву: "; // введення елементів масиву
    for (int i = 0; i < N; i++) {

```

```
        cin >> A[i];
    }

    cout << "Початковий масив: "; // виведення початкового масиву
    for (int i = 0; i < N; i++) {
        cout << A[i] << " ";
    }
    cout << endl;

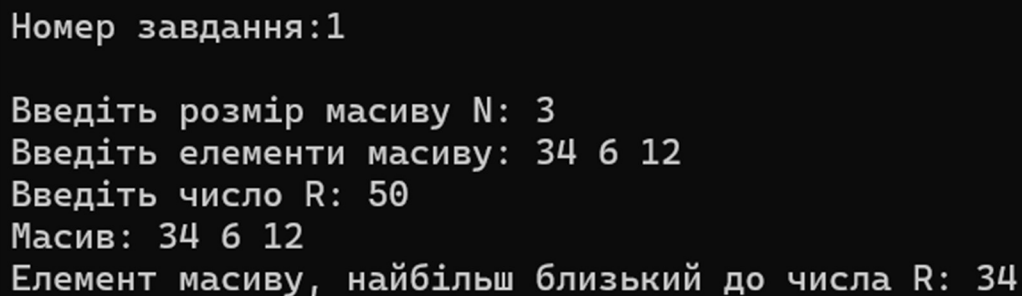
    // зрушення елементів вправо на одну позицію
    for (int i = N - 1; i > 0; i--) {
        A[i] = A[i - 1]; // переміщення кожного елементу на наступну позицію
        вправо
    }

    A[0] = 0; // перший елемент масиву стає рівним 0

    cout << "Змінений масив: "; // виведення зміненого масиву
    for (int i = 0; i < N; i++) {
        cout << A[i] << " ";
    }
    cout << endl;

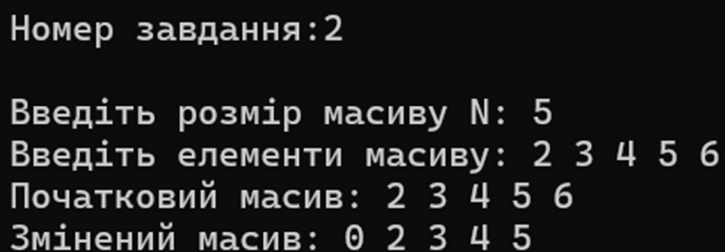
    delete[] A; // звільнення пам'ять після використання масиву
}
```

ДОДАТОК Б
Скрін-шоти вікна виконання програми



```
Номер завдання:1  
  
Введіть розмір масиву N: 3  
Введіть елементи масиву: 34 6 12  
Введіть число R: 50  
Масив: 34 6 12  
Елемент масиву, найбільш близький до числа R: 34
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Array 40



```
Номер завдання:2  
  
Введіть розмір масиву N: 5  
Введіть елементи масиву: 2 3 4 5 6  
Початковий масив: 2 3 4 5 6  
Змінений масив: 0 2 3 4 5
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Array 79

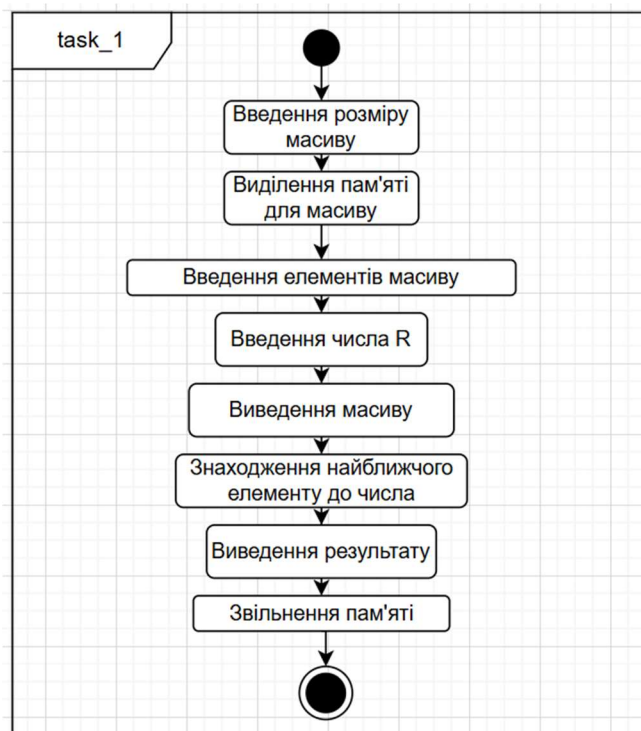


Рисунок Б.3 – Діаграма для завдання Array 40

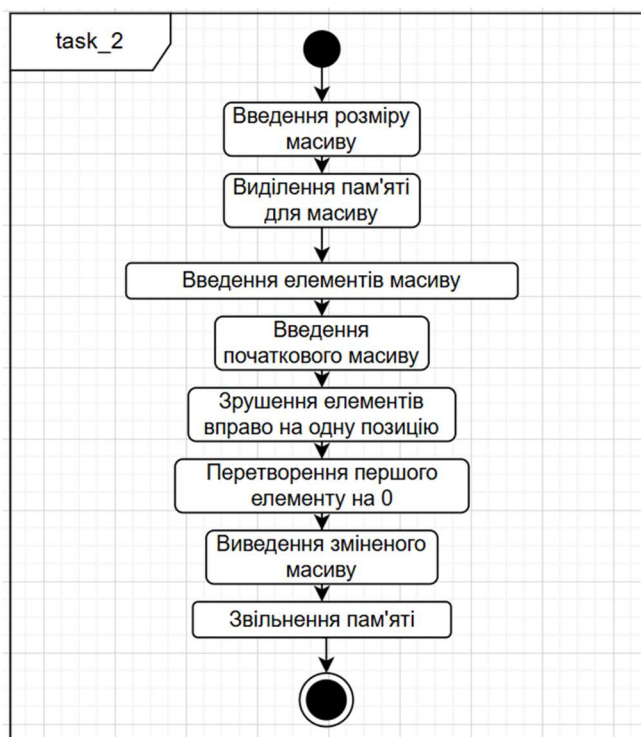


Рисунок Б.4 – Діаграма для завдання Array 79

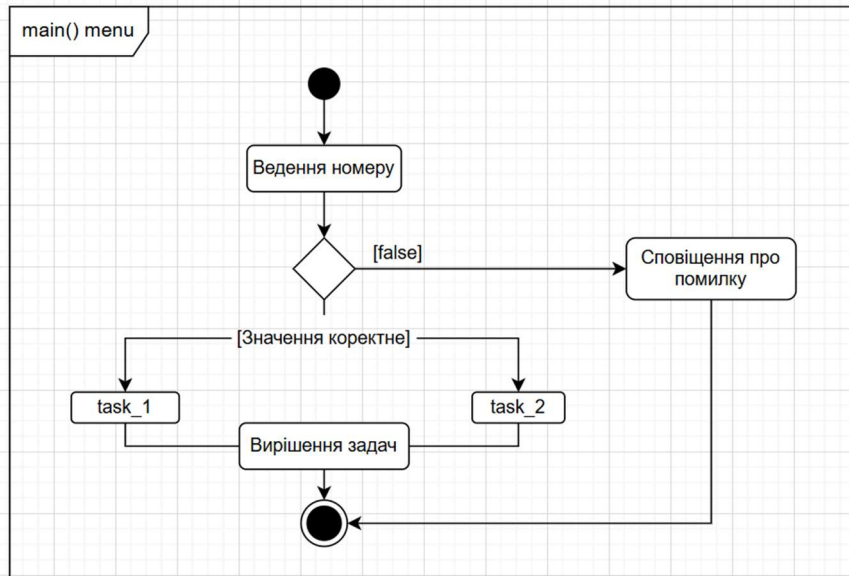


Рисунок Б.5 – Діаграма для завдання 3