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 № 2**

з дисципліни «Алгоритмізація та програмування»

на тему «"Математичні обчислення на мові С ++"»

Виконав студент гр. \_\_\_\_\_\_319\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Микита Тiтов\_\_\_\_\_\_\_*

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_ к.т.н., доц. Олена  ГАВРИЛЕНКО

(підпис, дата) (П.І.Б.)

2023

# МЕТА РОБОТИ

Вивчити теоретично базові типи даних мови С ++ і реалізувати консольний додаток лінійної структури для введення / виведення і обробки змінних базових типів з використанням вбудованих операцій та бібліотечних функцій на мові програмування C ++.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити задачу з цілочисельними змінними. Всі вхідні і вихідні дані в задачах цієї групи є цілими числами. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число і т. д.), вважаються додатними

Завдання 2. Вирішити завдання з логічними змінними. У всіх завданнях даної групи потрібно вивести логічне значення true (1), якщо наведене висловлювання для запропонованих вхідних даних є істинним, і значення false (0) в іншому випадку. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число і т. д.), вважаються цілими додатними.

Завдання 3. Обчислити математичний вираз зі змінними дійсного типу, використовуючи стандартну бібліотеку cmath. Число π має бути визначено як константа дійсного типу. Вирази представлено в табл.3

# ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Integer15

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

Дано тризначне число. Вивести число, отримане при перестановці

цифр сотень і десятків вихідного числа (наприклад, 123 перейде в

213).

Вихідні дані (ім’я, опис, тип):

Змінні:

num:

Тип: int (ціле число).

Використовується для введення тризначного числа, над яким буде виконано перестановку цифр сотень і десятків.

absNum:

Тип: int.

Зберігає абсолютне значення числа num для роботи з позитивними значеннями.

hundreds:

Тип: int.

Зберігає значення цифри сотень числа num, отримане діленням на 100.

tens:

Тип: int.

Зберігає значення цифри десятків числа num, отримане за допомогою операцій ділення та залишку.

units:

Тип: int.

Зберігає значення цифри одиниць числа num через залишок від ділення на 10.

newAbsNum:

Тип: int.

Нове число після перестановки цифр сотень і десятків.

A, B, C:

Тип: int.

Три числа, введені користувачем для перевірки умови "подвійної нерівності" (A < B < C).

Типи даних:

int:

Тип даних для цілих чисел.

Використовується для змінних num, absNum, hundreds, tens, units, A, B, C, і повернення результатів функцій.

bool:

Логічний тип даних (істинність/хибність).

Використовується у функції isBetween для перевірки умови.

Текстові сповіщення:

std::cerr << "Введіть тризначне число" << std::endl;:

Сповіщення для користувача у випадку, якщо введене число не є тризначним.

Виводиться в стандартний потік помилок (std::cerr).

std::cout << "Число після перестановки цифр сотень і десятків: " << result << std::endl;:

Виводить результат першої задачі — число після перестановки цифр.

std::cout << "Число B знаходиться між числами A і C." << std::endl;:

Інформує користувача, що число B знаходиться між числами A та C.

std::cout << "Число B не знаходиться між числами A і C." << std::endl;:

Інформує користувача, що число B не відповідає умові подвійної нерівності.

Інші заголовки:

std::cout << "=== Задача 1: Перестановка сотень і десятків ===" << std::endl;

std::cout << "=== Задача 2: Перевірка подвійної нерівності ===" << std::endl;

Використовуються для зручності читання, щоб розділити задачі у виводі програми.

Робота функцій:

swapHundredsAndTens(int num):

Перевіряє, чи є число тризначним.

Розділяє число на сотні, десятки та одиниці.

Переставляє місцями сотні і десятки, повертає нове значення (з урахуванням знака).

isBetween(int A, int B, int C):

Перевіряє, чи знаходиться число B між A і C (у будь-якому порядку).

Алгоритм вирішення

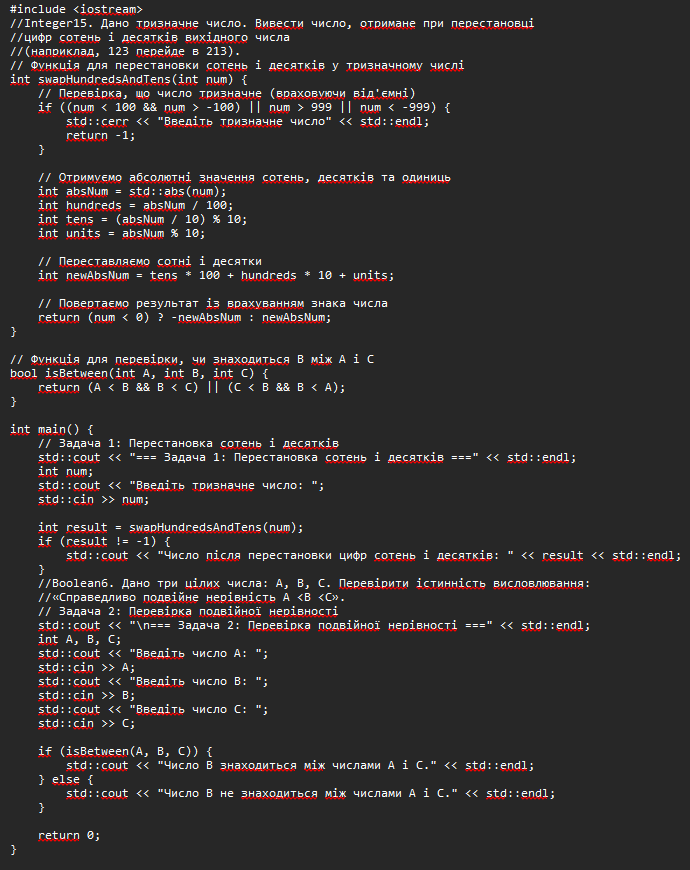


Рисунок 1 – Алгоритм вирiшення

Лістинг коду вирішення задачі:

#include <iostream>

//Integer15. Дано тризначне число. Вивести число, отримане при перестановці

//цифр сотень і десятків вихідного числа

//(наприклад, 123 перейде в 213).

// Функція для перестановки сотень і десятків у тризначному числі

int swapHundredsAndTens(int num) {

// Перевірка, що число тризначне (враховуючи від'ємні)

if ((num < 100 && num > -100) || num > 999 || num < -999) {

std::cerr << "Введіть тризначне число" << std::endl;

return -1;

}

// Отримуємо абсолютні значення сотень, десятків та одиниць

int absNum = std::abs(num);

int hundreds = absNum / 100;

int tens = (absNum / 10) % 10;

int units = absNum % 10;

// Переставляємо сотні і десятки

int newAbsNum = tens \* 100 + hundreds \* 10 + units;

// Повертаємо результат із врахуванням знака числа

return (num < 0) ? -newAbsNum : newAbsNum;

}

// Функція для перевірки, чи знаходиться B між A і C

bool isBetween(int A, int B, int C) {

return (A < B && B < C) || (C < B && B < A);

}

int main() {

// Задача 1: Перестановка сотень і десятків

std::cout << "=== Задача 1: Перестановка сотень і десятків ===" << std::endl;

int num;

std::cout << "Введіть тризначне число: ";

std::cin >> num;

int result = swapHundredsAndTens(num);

if (result != -1) {

std::cout << "Число після перестановки цифр сотень і десятків: " << result << std::endl;

}

//Boolean6. Дано три цілих числа: A, B, C. Перевірити істинність висловлювання:

//«Справедливо подвійне нерівність A <B <C».

// Задача 2: Перевірка подвійної нерівності

std::cout << "\n=== Задача 2: Перевірка подвійної нерівності ===" << std::endl;

int A, B, C;

std::cout << "Введіть число A: ";

std::cin >> A;

std::cout << "Введіть число B: ";

std::cin >> B;

std::cout << "Введіть число C: ";

std::cin >> C;

if (isBetween(A, B, C)) {

std::cout << "Число B знаходиться між числами A і C." << std::endl;

} else {

std::cout << "Число B не знаходиться між числами A і C." << std::endl;

}

return 0;

}

Екран роботи програми показаний на рис.

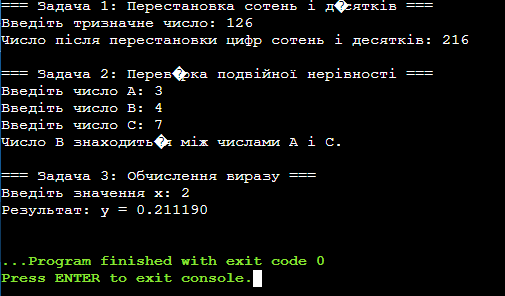


Рисунок 2. Екран роботи програми

# ВИСНОВКИ

Пiд час роботи вивчили теоретично базові типи даних мови С ++ і реалізували консольний додаток лінійної структури для введення / виведення і обробки змінних базових типів з використанням вбудованих операцій та бібліотечних функцій на мові програмування C ++.

ДОДАТОК А

Лістинг коду програми

#include <iostream>

//Integer15. Дано тризначне число. Вивести число, отримане при перестановці

//цифр сотень і десятків вихідного числа

//(наприклад, 123 перейде в 213).

// Функція для перестановки сотень і десятків у тризначному числі

int swapHundredsAndTens(int num) {

// Перевірка, що число тризначне (враховуючи від'ємні)

if ((num < 100 && num > -100) || num > 999 || num < -999) {

std::cerr << "Введіть тризначне число" << std::endl;

return -1;

}

// Отримуємо абсолютні значення сотень, десятків та одиниць

int absNum = std::abs(num);

int hundreds = absNum / 100;

int tens = (absNum / 10) % 10;

int units = absNum % 10;

// Переставляємо сотні і десятки

int newAbsNum = tens \* 100 + hundreds \* 10 + units;

// Повертаємо результат із врахуванням знака числа

return (num < 0) ? -newAbsNum : newAbsNum;

}

// Функція для перевірки, чи знаходиться B між A і C

bool isBetween(int A, int B, int C) {

return (A < B && B < C) || (C < B && B < A);

}

int main() {

// Задача 1: Перестановка сотень і десятків

std::cout << "=== Задача 1: Перестановка сотень і десятків ===" << std::endl;

int num;

std::cout << "Введіть тризначне число: ";

std::cin >> num;

int result = swapHundredsAndTens(num);

if (result != -1) {

std::cout << "Число після перестановки цифр сотень і десятків: " << result << std::endl;

}

//Boolean6. Дано три цілих числа: A, B, C. Перевірити істинність висловлювання:

//«Справедливо подвійне нерівність A <B <C».

// Задача 2: Перевірка подвійної нерівності

std::cout << "\n=== Задача 2: Перевірка подвійної нерівності ===" << std::endl;

int A, B, C;

std::cout << "Введіть число A: ";

std::cin >> A;

std::cout << "Введіть число B: ";

std::cin >> B;

std::cout << "Введіть число C: ";

std::cin >> C;

if (isBetween(A, B, C)) {

std::cout << "Число B знаходиться між числами A і C." << std::endl;

} else {

std::cout << "Число B не знаходиться між числами A і C." << std::endl;

}

return 0;

}

ДОДАТОК Б

Скрін-шоти вікна виконання програми

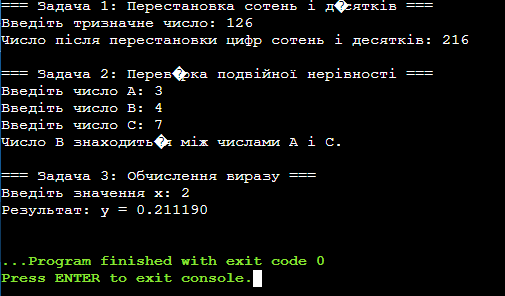


Рисунок Б.1 – Екран виконання програми для вирішення завдання