 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №3**

з дисципліни «Алгоритмізація та програмування»

на тему: «Алгоритми послідовної (лінійної) структури»

Варіант №17

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

доцент каф. БМК

к.т.н. Алхімова С.М.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2019

**Завдання:**

1. Вивчити теоретичні основи написання алгоритмів послідовної (лінійної) структури.
2. Побудувати блок-схему алгоритму вирішення завдання.
3. Відповідно до свого варіанту написати програму, яка розраховує значення виразу за заданими в коді значеннями змінних **a, b, c** і **d**.

**a = 0.6, b = 0.2, c = 1.1, d = -2.5**

1. **Додаткове завдання\*.** Внести зміни до коду основного завдання, за рахунок яких значення змінних **a, b, c** і **d** вводяться користувачем з клавіатури. Для введення даних з клавіатури та виведення даних на екран використовувати функції форматованого введення-виведення. Побудувати блок-схему алгоритму вирішення додаткового завдання.
2. Скласти і захистити звіт по роботі.

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок-схема:**



**Код:**

#include <stdio.h> // заголовочный файл input/output

#include <math.h> // заголовочный файл для математических функций

#include <conio.h> // консольный input/output

int main()

{

float a = 0, b = 0, c = 0, d = 0; // оглашение переменных

printf("Enter a,b,c,d please:\n");

scanf\_s("%f %f %f %f", &a, &b, &c, &d); // ввод четырех значений

float y = sqrt(log(c \* fabs(2 \* d))) + 5 \* (cos(a) / acos(-2 \* b)); // инициализация y выражением

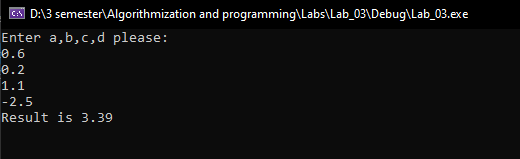
printf("Result is %.2f", y); // вывод y

\_getch();

return 0;

}

**Результати:**



**Контрольні питання:**

1. Що таке алгоритм послідовної ( лінійної) структури, програма послідовної ( лінійної) структури?

Алгоритм послідовної ( лінійної) структури- це алгоритм, в якому блоки виконуються послідовно один за одним в порядку, що заданий схемою.

Програма послідовної ( лінійної) структури- найпростіша програма використовується для реалізації звичайних обчислень за формулами.

1. Дайте визначення оператора, операнди та операції.

Операція — дії, що виконується над елементами операторів.

Операнд — аргумент операції; дані, які обробляються командою; граматична конструкція, яка позначає вираз, що задає значення аргументу операції; іноді операндом називають місце або позицію в тексті, де має стояти аргумент операції.

Оператор — це конструкція для опису деякої дії в процесі обробки даних (наприклад, +,-,%, <<).

1. Перелічити основні стандартні типи даних визначені в мові С++?

char, unsigned char, int, unsigned int, short, unsigned short, long, unsigned long, float, double, long double.

1. Що таке операції декремента, інкремента?

Операція інкремента - операція, що збільшує змінну на 1 (наприклад i++/++i).

Операція декремент - зворотня операція інкременту, тобто операція, що зменшує змінну на 1 (наприклад i--/--i).

1. Як визначають коментарі у програмному коді? Навести відповідні приклади.

Коментарі в програмі не впливають на компіляцію програми, а слугують для роз’яснення основних блоків тексту і програми в цілому.

Можна виділити :

* Однорядковий коментар, активний від двох символів // до кінця рядка.

Наприклад: «cin >> a; // input of variable»

* Багаторядковий коментар визначаеться всередині пари символів /\*і\*/.

Наприклад: « using namespace std;

/\*--------- main function ----------\*/

int main() »

1. Яка різниця під час використання форматованого та потокового введення-виведення даних?

На відміну від відповідних конструкцій форматного введення-виведення, потокові оператори не вимагають явного вказування типу змінних. Це відбувається за рахунок автоматичного визначення типу змінної в потокових операціях.

1. Що визначає ширина поля вводу/виводу (навести відповідні приклади)?

При вводі, ширина поля - позитивне десяткове ціле число, яке задає максимальне число символів, що може бути прочитане з вхідного потоку, доки не зустрінеться символ « » (пропуск) або символ, який не може бути перетворений відповідно до заданого формату.

При виводі - визначає мінімальну кількість виведених символів, якщо довжина більше виведеної кількості символів, то виведене значення доповнюється пропусками, у випадку, коли довжина менше кількості символів у виведеному значенні або вона не задана, виводяться всі символи значення (відповідно до поля точність, якщо воно є).

При потоковому виводі ширина поля визначається лиши для даних символьного типу.

Приклади:

cin.setf(ios::left);

cin.width(13);

cin >> a; // cpp1234567890end -> cpp1234567890

cout.width(13);

cout << «cppstudio.com» << endl; // \_\_\_\_\_\_\_\_\_\_cpp

scanf\_s("%5d", &b); // 123456 -> 12345

printf("%20s\n", "this is a test");// \_\_\_\_\_\_this is a test

1. Як визначити максимальну кількість цифр після десяткової точки для виведення на екран даних дійсного типу?

Для цього існуе точність - задається цілим числом після крапки і визначає кількість виведених символів, кількість знаків після крапки; на відміну від поля довжини поле точність може привести до «врізання» виведених даних.

Для потокового виводу даних маємо setprecision(n), де n кількість цифр після десяткової точки.

Приклади:

std::cout << std::fixed;

std::cout << std::setprecision(5) << 1.2342254 << '\n'; // 1.23422

printf(“%.5f”, 1.2342254); //1.23422

1. Для чого необхідні стандартні файли заголовків?

Використання текстових файлів заголовків (header-файлів), які вставляються в текст програми на мові С++ з допомогою директиви include, є традиційною технікою програмування на мові С++, що забезпечує синтаксичну правильність використання бібліотечних функцій (в тому числі і системних викликів) в прикладній програмі. Тобто стандартні файли заголовків, до яких звертається компілятор, необхідні для опису використовуваних функцій в програмі, які необхідні користувачу конкретно в даній програмі. Наприклад у цій лабораторній було використано 3 стандартні файли заголовків:

#include <stdio.h> // заголовочный файл input/output

#include <math.h> // заголовочный файл для математических функций

#include <conio.h> // консольный input/output

1. Як зміниться значення змінної i в рядку програми i=i+1 ?

Значення змінної i збільшиться на 1.