**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут: **ІКНІ**

Кафедра: **Пз**

**Звіт**

До лабораторної роботи №1

**На тему:** «Основні поняття мови С реалізація

алгоритмів з розгалуженням»

**З дисципліни:** «Основи програмування»

**Лектор:**

ст. викл. каф. ПЗ

Муха.Т.О

**Виконав:**

Ст. гр. ПЗ-15

Бабіля О.О.

**Прийняла:**

Асист. каф. Пз

Заводовська Н.О

13.09.2021

∑= \_\_\_\_ .

Львів – 2021

**Тема** **роботи**: написання найпростіших лінійних алгоритмів з розгалуженням на мові С

**Мета** **роботи**: навчитися програмувати на мові С найпростіші лінійні алгоритми та алгоритми з галуженням

**Індивідуальне завдання 1-1**

1.Написати програму для обчислення заданих арифметичних виразів. Вважати, що X, Y – змінні, значення яких слід вводити з клавіатури, всі інші величини виразу описати як константи.



**Індивідуальне завдання 2-2**

1.Не використовуючи оператор циклу скласти програму, яка

**Теоритичні відомості**

## **Ідентифікатори**

Ідентифікатор — це ім’я змінної, функції, класу або іншого об’єкту в C. Ми можемо визначати ідентифікатори будь-якими словами/іменами. Тим не менш, є кілька загальних правил, яких необхідно дотримуватися:

Правило №1: Ідентифікатор не може бути ключовим словом. Ключові слова зарезервовані.

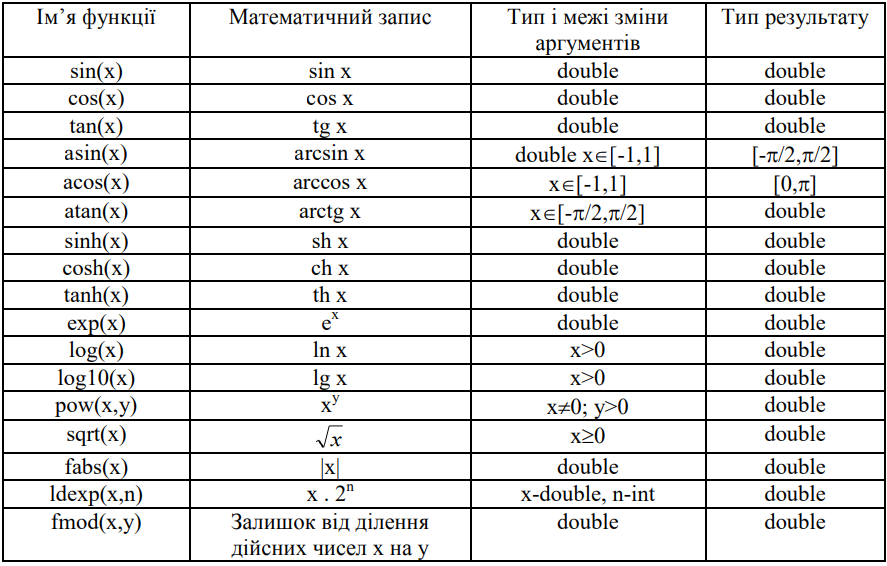
Правило №2: Ідентифікатор може складатися тільки з букв (нижнього або верхнього регістру), цифр або символів підкреслення. Це означає, що всі інші символи та пробіли заборонені.

Правило №3: Ідентифікатор повинен починатися з літери (нижнього або верхнього регістру). Він не повинен починатися з цифри.

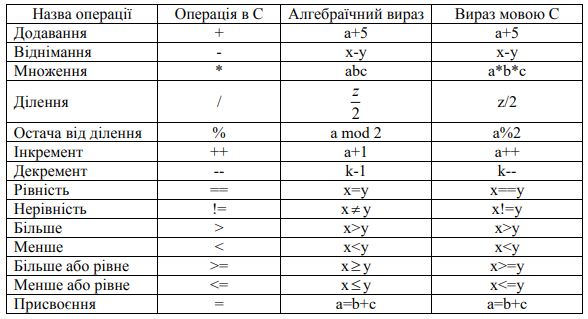
Правило №4: Мова C розрізняє нижній регістр від верхнього Nvalue відрізняється від nValue і відрізняється від NVALUE.

**Функції**

Програма складається з сукупності фунцій, одна з яких повина називатися main(). Опис функції складається з заголовка і тіла функції. Заголовок містить оператори(директиви) препроцесора, наприклад, #include і ім'я функції. Тіло функції розташоване у фігурних дужках і складається з ряду операторів, кожен з яких закінчується крапкою з комою.

**Основні математичні функції в С:**

**Елементарні** **опреації** **мови** **С** :



**Оператор** **умови** **if**

Конструкція Ifдозволяє контролювати чи слід програмі виконувати певну частину коду чи ні в залежності від того якою є дана умова — істинною або хибною. Однією з важливих властивостей конструкції Ifє те, що вона дозволяє програмі вибрати певну дію, ґрунтуючись на введені користувачем дані .Оператор умови задає певній дії твердження істинне(True) або хибне(False) . Істинне твердження є одиницею, результатом якого є відмінне від нуля число. Хибне твердження оцінюється як нуль. При виконанні порівняння з реляційними операторами, оператор повертає 1, якщо порівняння істинно, або 0, якщо порівняння невірно. Наприклад, перевірка 0 == 2 повертає 0. Перевірка 2 == 2 оцінюється в 1.

Будова оператора if: **if (<умова>)<оператор> ;**

**Код програми 1**

**#include <stdio.h>**

#include <math.h>

void main()

{

const int z = -2 ;

double x, y, r, squarex, squarey, squarez;

printf("Enter x,y:");

scanf\_s("%lf %lf", &x, &y);

squarex = x \* x;

squarey = y \* y;

squarez = z \* z;

r = sqrt(squarex + squarey + squarez);

printf("r=%lf", r);

return 0;

**}**

**Результат виконання програми**

**1)5.2 \* 5.2 = 27.04**

**2)6\*6=36**

**3)-2\*-2=4**

**4)27.04+6\*6+4=67.04**

**5)**√ **67.04=8,187796**

**Код програми 2**

**# include <stdio.h>**

void main()

{

double x,z;

int y;

printf("Enter x:");

scanf\_s("%lf", &x);

if (x >= 0);

{

y = x / 360;

z = x - 360 \* y;

if (z > 0 && z < 180) printf("sinx = positive\n");

if (z > 180 && z < 360) printf("sinx = negative\n");

if (z > 0 && z < 90) printf("cosx = positive\n");

if (z > 270) printf("cosx = positive\n");

if (z > 90 && z < 270) printf("cosx = negative\n");

if (z > 0 && z < 90) printf("tgx = positive\n");

if (z > 0 && z < 90) printf("ctgx = positive\n");

if (z > 180 && z < 270) printf("tgx = positive\n");

if (z > 180 && z < 270) printf("ctgx = positive\n");

if (z > 90 && z < 180) printf("tgx = negative\n");

if (z > 90 && z < 180) printf("ctgx = negative\n");

if (z > 270) printf("tgx = negative\n");

if (z > 270) printf("ctgx = negative\n");

if (z == 0) printf("sinx = 0\n");

if (z == 0) printf("cosx = positive\n");

if (z == 0) printf("tgx = 0\n");

if (z == 0) printf("ctgx = does not exist\n");

if (z == 90) printf("cosx = 0\n");

if (z == 90) printf("tgx = does not exist\n");

if (z == 90) printf("ctgx = 0\n");

if (z == 180) printf("sinx = 0\n");

if (z == 180) printf("tgx = 0\n");

if (z == 180) printf("ctgx = does not exist\n");

if (z == 270) printf("cosx = 0\n");

if (z == 270) printf("tgx = does not exist\n");

if (z == 270) printf("ctgx = 0\n");

}

if (x < 0)

{

y = x / 360;

z = x - 360 \* y;

if (z < 0 && z > -180) printf("sinx = negative\n");

if (z < -180) printf("sinx = positive\n");

if (z < 0 && z > -90) printf("cosx = positive\n");

if (z < -90 && z > -270) printf("cosx = negative\n");

if (z < -270) printf("cosx = positive\n");

if (z < 0 && z > -90) printf("tgx = negative\n");

if (z < 0 && z > -90) printf("ctgx = negative\n");

if (z < -180 && z > -270) printf("tgx = negative\n");

if (z < -180 && z > -270) printf("ctgx = negative\n");

if (z < -90 && z > -180) printf("tgx = positive\n");

if (z < -90 && z > -180) printf("ctgx = positive\n");

if (z < -270) printf("tgx = positive\n");

if (z < -270) printf("ctgx = positive\n");

if (z == -270) printf("cosx = 0\n");

if (z == -270) printf("tgx = does not exist\n");

if (z == -270) printf("ctgx = 0\n");

if (z == -180) printf("sinx = 0\n");

if (z == -180) printf("tgx = 0\n");

if (z == -180) printf("ctgx = does not exist\n");

if (z == -90) printf("cosx = 0\n");

if (z == -90) printf("tgx = does not exist\n");

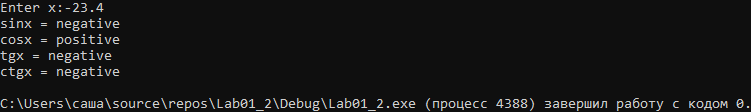
if (z == -90) printf("ctgx = 0\n");

}

return 0;

}

**Результат виконання програми 2**

****

****

**Висновки:** В ході даної лабораторної роботи я удосконалив свої вміння у програмуванні найпростіших

лінійних алгоритмів та алгоритмів з галуженням на мові С. Набув теоретичних та практичних знань

у програмуванні на мові С.