Integer5

#include <iostream>

using namespace std;

int main() {

int A, B;

// Integer5.Дано цілі додатні числа A і B (A> B).

// На відрізку довжиною A розміщено максимально можлива кількість відрізків довжиною B(без накладання).

// Використовуючи операцію взяття залишку від ділення,знайти довжину незайнятої частини відрізка A.

// Input values for A and B

cout << "Enter the value of A (A > B): ";

cin >> A;

cout << "Enter the value of B: ";

cin >> B;

// Check if A > B condition is met

if (A > B) {

int unused\_part = A % B;

cout << "Length of the unused part of segment A: " << unused\_part << endl;

}

else {

cout << "Error: A should be greater than B." << endl;

}

return 0;

}

Boolean26.

#include <iostream>

using namespace std;

int main() {

//Дано числа x, y.

//Перевірити істинність висловлювання: «Точка з координатами(x, y) лежить в четвертій координатної чверті».

double x, y;

// Input coordinates x and y

cout << "Enter the x-coordinate: ";

cin >> x;

cout << "Enter the y-coordinate: ";

cin >> y;

// Check if the point is in the fourth quadrant

bool isInFourthQuadrant = (x > 0 && y < 0);

if (isInFourthQuadrant) {

cout << "The point (" << x << ", " << y << ") is in the fourth quadrant." << endl;

}

else {

cout << "The point (" << x << ", " << y << ") is not in the fourth quadrant." << endl;

}

return 0;

}

Таблиця 3 – Математичні вирази №2

#include <iostream>

#include <cmath>

using namespace std;

int main() {

// Я не знаю як встати сюди текст завдання

const double PI = 3.141592653589793;

double x;

// Input x

cout << "Enter the value of x: ";

cin >> x;

// Calculate each part of the expression

double exponent\_part = exp(x + 0.5);

double tangent\_part = tan(x + 13 \* PI / 180); // Convert 13 degrees to radians

double square\_root\_part = sqrt(fabs(x - tangent\_part + 25));

double sine\_part = pow(sin(pow(x, 3)), 2);

double log\_part = log(fabs(x)) / log(5); // Change of base to get log base 5

double denominator = cbrt(sine\_part \* pow(x, 3) \* log\_part);

// Check if the denominator is zero to avoid division by zero

if (denominator == 0) {

cout << "Error: Division by zero in the denominator." << endl;

}

else {

double y = (exponent\_part \* square\_root\_part) / denominator;

cout << "The result y is: " << y << endl;

}

return 0;

}