**­**

**Лабораторна робота №3**

З дисципліни «Програмування»

Тема: ПОБУДОВА ГРАФІКА ФУНКЦІЇ

Виконав: Землянський Едуард

Група: КВ-22

**Постановка задачі**

Використовуючи графічний режим, побудувати графік заданої функції, розв’язуючи наступні підзадачі:

1. Можливість зміни інтервалу обчислення значень функції (тобто кількість проміжків розбиття N задається константою).
2. Аналіз діапазону зміни значень функції на заданому проміжку зміни аргументу.
3. Зображення на екрані координатних осей X та Y з розміченою шкалою та проставленими числами, що відповідають діапазону зміни аргументу та функції.
4. Зображення на екрані графіка функції, формула якої визначається варіантом завдання.
5. Можливість масштабування виведеного зображення.
6. Зображення на вільній ділянці екрану математичної формули а також прізвища студента, що виконав завдання.

**Вимоги до програми**

Графік функції доцільно зображувати у вигляді ламаної лінії, що з’єднує точки, які належать графіку, тобто за допомогою частково-лінійної апроксимації кривої графіка.

Для побудови графіка по заданому рівнянню необхідно обчислити масив значень аргументу та масив відповідних значень функції. Крок зміни аргументу залежить від обраного діапазону зміни аргументу й кількості точок і обчислюється за формулою:

, де

*dx* - крок зміни аргументу;

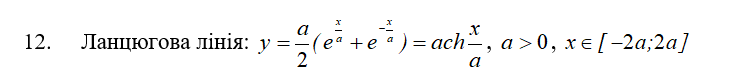
*xmax* , *xmin* - відповідно максимальне та мінімальне значення аргументу;

*num* - обрана кількість точок графіку.

Паралельно з обчисленням масивів значень аргументу  та функції  необхідно визначити мінімальне й максимальне значення функції, які будуть потрібні для проведення масштабування.

Для зображення графіка на екрані обирається прямокутна ділянка, що визначається координатами лівого верхнього кута  і правого нижнього кута . При цьому потрібно мати на увазі, що лівіше й нижче поля виводу будуть знаходитися написи (відповідні значення аргументу і функції).

**Варіант завдання**

****

**Текст програми**

#include <GL/freeglut.h>

#include <GL/glut.h>

#include <math.h>

#include <iostream>

#include <time.h>

float a = 1; // Параметр a

float x1, x2; // Відрізок на якому буде згенеровано графік

float scale = 1.0; // Масштаб

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0, 1.0, 1.0);

glRasterPos2f(-x2+0.5, x2-1.5);// Встановлення позиції растрового тексту

const char\* name = "Eduard Zemlyanski KV-22";

for (int i = 0; name[i] != '\0'; i++)

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, name[i]); // Виведення символу

}

glRasterPos2f(-x2+0.5, x2-2); // Встановлення позиції растрового тексту

const char\* equation1 = "y = arccosh(x / a);";

for (int i = 0; equation1[i] != '\0'; i++)

{

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, equation1[i]);

}

glLineWidth(4.0);

glBegin(GL\_LINE\_STRIP); // Зміна режиму малювання на GL\_LINE\_STRIP

glColor3f(1.0, 0.5, 0);

for (float x = x1; x <= x2; x += 0.01)

{

float y = a \* acosh(x / a);

glVertex2f(x \* scale, y \* scale);

}

glEnd();

glLineWidth(1.0);

// Відображення координатних вісей з підписами та маркерами

glColor3f(1.0, 1.0, 1.0);

glBegin(GL\_LINES);

// Вісь X

glVertex2f(-x2, 0.0);

glVertex2f(x2, 0.0);

for (float i = -x2; i <= x2; i += 1.0) {

glVertex2f(i\* scale, 0.1);

glVertex2f(i\* scale, -0.1);

glRasterPos2f(i\* scale, -0.3);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, '0' + (int)i);

}

// Вісь Y

glVertex2f(0.0, -x2);

glVertex2f(0.0, x2);

glRasterPos2f(0.3, x2\*scale);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, 24);

for (float i = -x2; i <= x2; i++) {

glVertex2f(0.1, i\*scale);

glVertex2f(-0.1, i\*scale);

glRasterPos2f(-0.3, i\*scale);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, (int)i + '0');

}

glEnd();

glFlush();

}

void reshape(int width, int height)

{

glViewport(0, 0, width, height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-x2, x2, -x2, x2, -1.0, 1.0);

glMatrixMode(GL\_MODELVIEW);

}

void mouseWheel(int wheel, int direction, int x, int y)

{

if (direction == 1)

{

// Zoom in

scale \*= 1.1;

}

else

{

// Zoom out

scale /= 1.1;

}

// Redraw the scene

glutPostRedisplay();

}

void SetData(){

std::cout << "Enter 'a': ";

scanf("%f", &a);

std::cout << "Enter 'x1': ";

scanf("%f", &x1);

std::cout << "Enter 'x2': ";

scanf("%f", &x2);

if (a < 0 || x2 <= x1){

std::cout << "Wrond data given";

Sleep(2000);

system("cls");

SetData();

}

}

int main(int argc, char\*\* argv)

{

SetData();

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(800, 800);

glutCreateWindow("arccosh(x/a) function graph");

glutMouseWheelFunc(mouseWheel);

glutDisplayFunc(display);

glutReshapeFunc(reshape);;

glClearColor(0.0, 0.0, 0.0, 1.0);

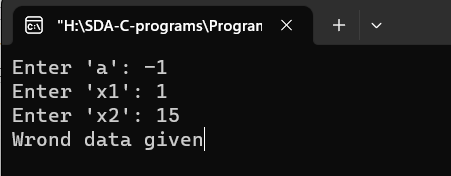
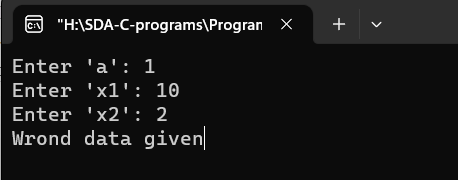
glutMainLoop();

return 0;

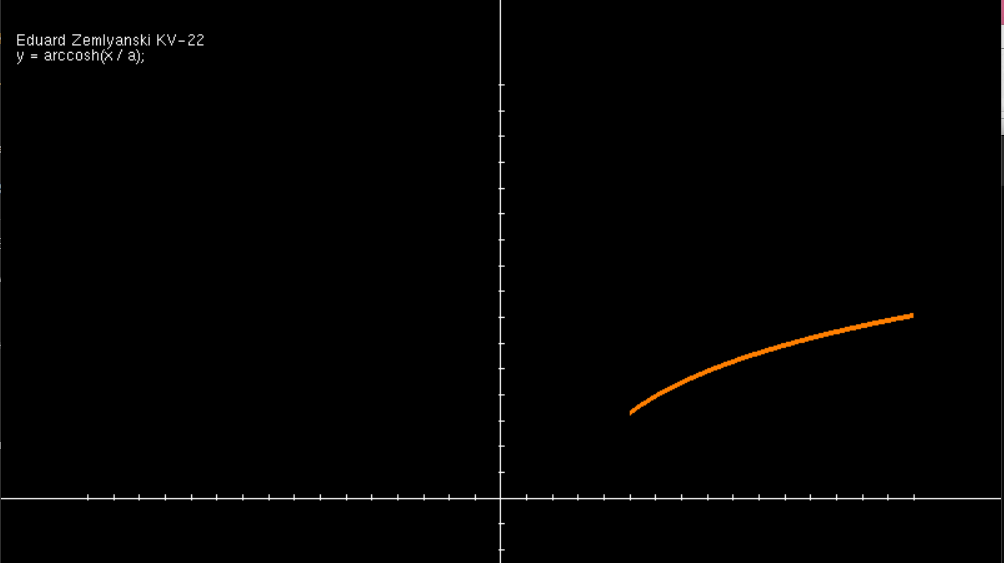
}

**Тестування програми**

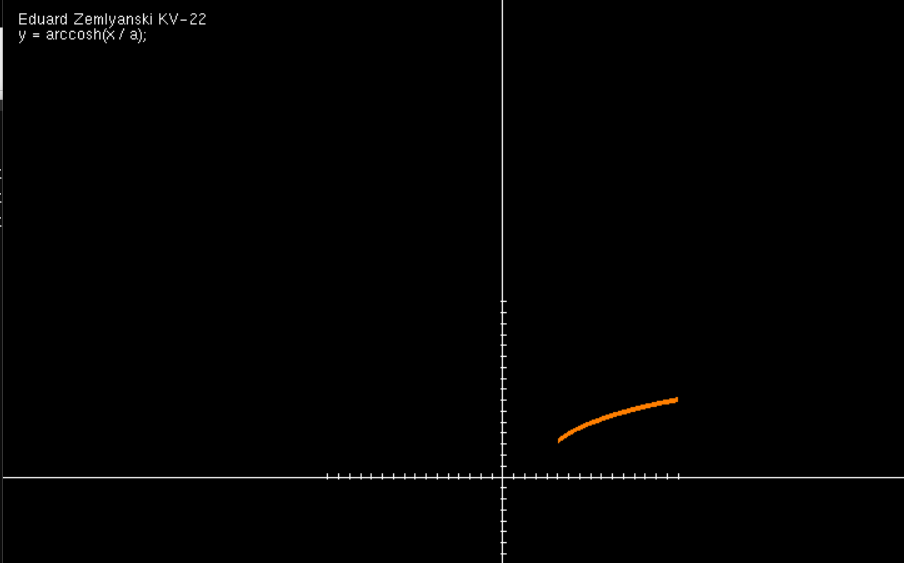
Програма проситиме ввід даних поки користувач не введе їх вірно (рекурсивна войд функція):

А=3, х1=5, х2=15



А=3, х1=5, х2=15 (значення ті самі, змінений масштаб прокручуванням колесика миші)



А=3, х1=-6, х2=25

