ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

за курсом “Програмування”

студента групи ПА-22-2

Овдієнка Андрія Володимировича

кафедра комп’ютерних технологій, ДНУ

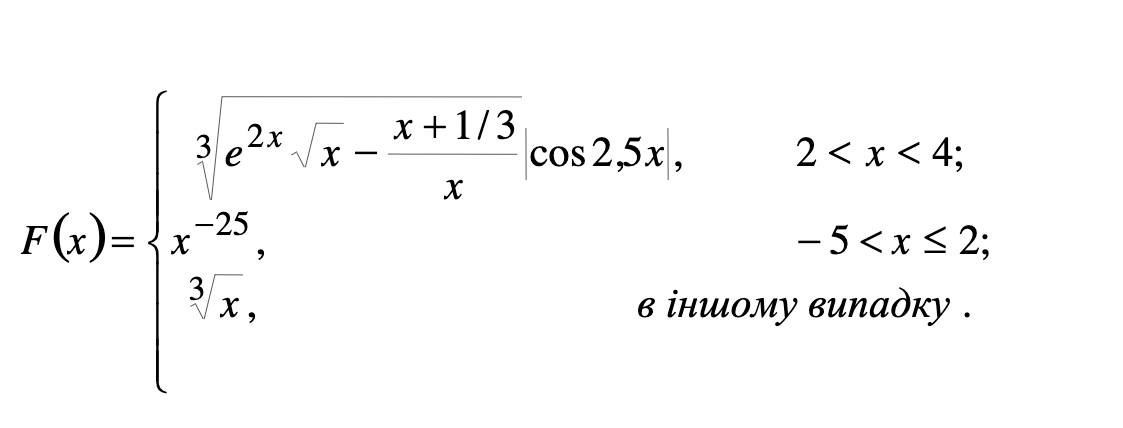
2022/2023

1.Завдання до лабораторної роботи №6.1

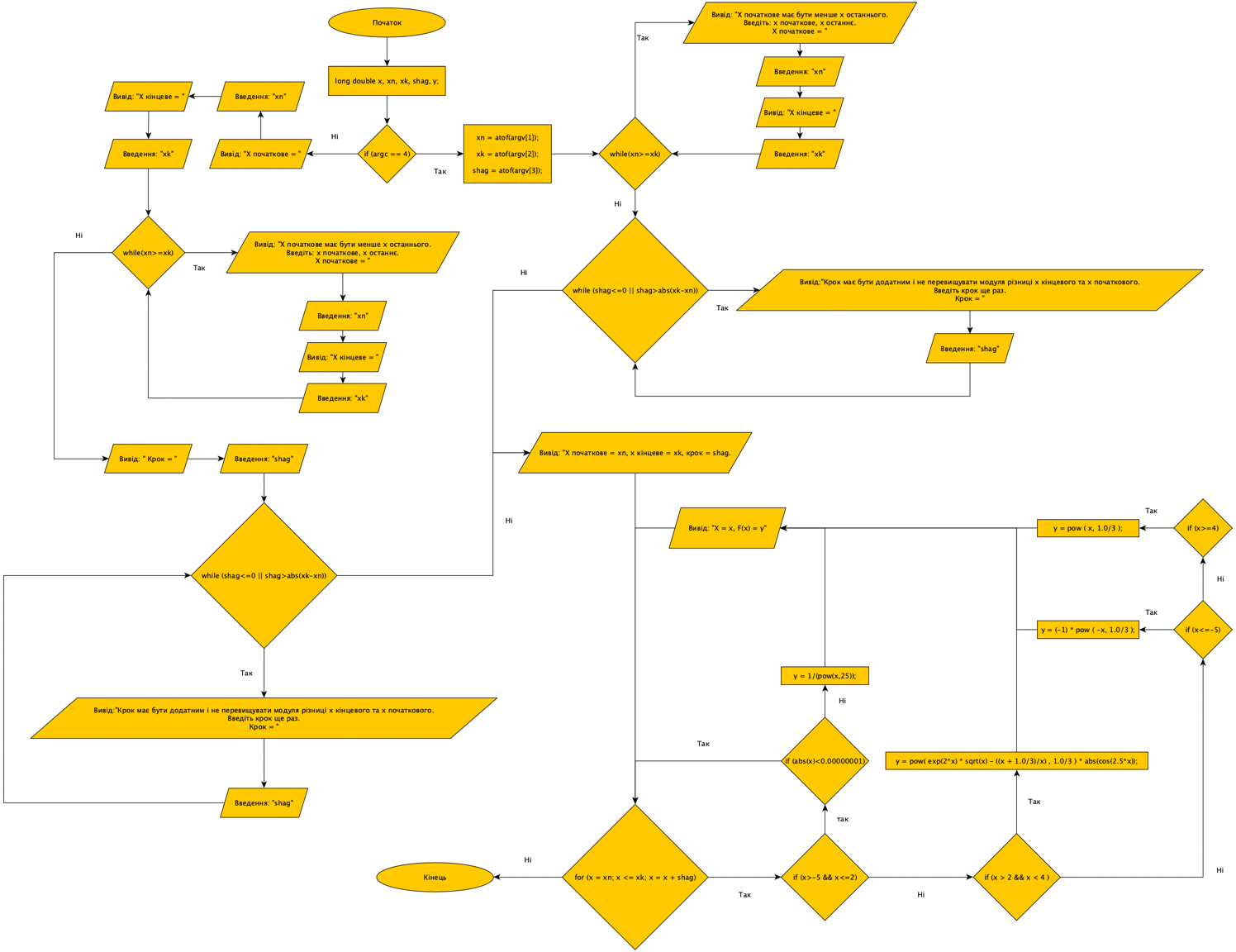
Постановка задачі:

Складіть програму, яка виводить на екран таблицю значень функції F(x) у заданому діапазоні. Значення функції F(x) обчисліть за формулою, наведеною у вашому варіанті індивідуального завдання. Визначте області допустимих значень параметрів формул (у разі непопадання значення х до ОДЗ – видавати відповідне повідомлення). Діапазон і крок зміни аргументу користувач задає в командному рядку, в тому разі, якщо в командному рядку параметри не задано або їх задано в недостатній кількості, програма пропонує здійснити ввід з клавіатури під час виконання програми. Передбачити перевірку правильності введених користувачем аргументів. Значення F(x) обчислюють в окремій функції.

Індивідуальне завдання:



2.Опис розв’язку (блок-схема)



3.Вихідний текст програми розвʼязку задачі (код програми з коментарями)

#include <iostream> // Підключаємо бібліотека для вводу й виводу символів на консоль.

#include <cmath> // Підключаємо бібліотека для використання математичних функцій.

using namespace std; // Вказуємо, який простір імен використовуємо.

int main(int argc, char \*argv[]) // Програма на С++ починається із виконання функції main, у якій ми передаємо 4 елементи в програму одразу з терміналу, преший з яких - назва.

{

cout << "\n"; // Перехід на новий рядок.

long double x, xn, xk, shag, y; // Оголошуємо змінні: значення "х", "х" початкове, "х" кінцеве, крок.

if (argc == 4) // Якщо передано 4 аргументи.

{

xn = atof(argv[1]); // "x" початкове приймає значення другого аргументу.

xk = atof(argv[2]); // "x" кінцеве приймає значення третього аргументу.

shag = atof(argv[3]); // Крок приймає значення четвертого аргументу.

while(xn>=xk) // Поки "x" початкове більше або дорівнює "х" кінцевому.

{

cout << "Х початкове має бути менше х останнього.\nВведіть: x початкове, x останнє.\nX початкове = "; // Виводимо на консоль символи, пишемо користувачу, що "х" початкове має бути менше "х" останнього, переходимо на новий рядо,просимо ввести "х" початкове та "х" кінцеве, переходимо на новий рядок, запитуємо значення "х" початкового.

cin >> xn; // Користувач вводить "х" початкове.

cout << "X кінцеве = "; // Виводимо на консоль символи, просимо користувача ввести "х" кінцеве.

cin >> xk; // Користувач вводить "х" кінцеве.

}

while (shag<=0 || shag>abs(xk-xn)) // Поки крок менш або дорівнює нулю; або більше модуля різниці "х" кінцевого та "х" початкового.

{

cout << "Крок має бути додатним і не перевищувати модуля різниці х кінцевого та х початкового.\nВведіть крок ще раз.\nКрок = "; // Виводимо на консоль символи, пишемо користувачу, що крок має бути додатним і не перевищувати модуля різниці "х" кінцевого та "х" початкового, переходимо на новий рядо, просимо ввести крок ще раз, переходимо на новий рядок, запитуємо значення кроку.

cin >> shag; // Користувач вводить крок.

}

}

else // Інакше.

{

cout << "X початкове = "; // Виводимо на консоль символи, просимо користувача ввести "х" початкове.

cin >> xn; // Користувач вводить "х" початкове.

cout << "X кінцеве = "; // Виводимо на консоль символи, просимо користувача ввести "х" кінцеве.

cin >> xk; // Користувач вводить "х" кінцеве.

while(xn>=xk) // Поки "x" початкове більше або дорівнює "х" кінцевому.

{

cout << "Х початкове має бути менше х останнього.\nВведіть: x початкове, x останнє.\nX початкове = "; // Виводимо на консоль символи, пишемо користувачу, що "х" початкове має бути менше "х" останнього, переходимо на новий рядо,просимо ввести "х" початкове та "х" кінцеве, переходимо на новий рядок, запитуємо значення "х" початкового.

cin >> xn; // Користувач вводить "х" початкове.

cout << "X кінцеве = "; // Виводимо на консоль символи, просимо користувача ввести "х" кінцеве.

cin >> xk; // Користувач вводить "х" кінцеве.

}

cout << "Крок = "; // Виводимо на консоль символи, просимо користувача ввести крок.

cin >> shag; // Користувач вводить крок.

while(shag<=0 || shag>abs(xk-xn)) // Поки крок менш або дорівнює нулю; або більше модуля різниці "х" кінцевого та "х" початкового.

{

cout << "Крок має бути додатним і не перевищувати модуля різниці х кінцевого та х початкового.\nВведіть крок ще раз.\nКрок = "; // Виводимо на консоль символи, пишемо користувачу, що крок має бути додатним і не перевищувати модуля різниці "х" кінцевого та "х" початкового, переходимо на новий рядо, просимо ввести крок ще раз, переходимо на новий рядок, запитуємо значення кроку.

cin >> shag; // Користувач вводить крок.

}

}

cout << "\nX початкове = " << xn << ", x кінцеве = " << xk << ", крок = " << shag << ".\n\n"; // Переходимо на новий рядок, виводимо числа, якому дорівнюють значення "х" початкового, "х" останнього та кроку, переходимо двічі на новий рядок.

for (x = xn; x <= xk; x = x + shag) // Під час того як "х" приймає значення "х" початкового, "х" менше або дорівнює "х" кінцевому, "x" приймає своє значення "х" та додається крок.

{

if (x>-5 && x<=2) // Якщо "х" строго більше ніж мінус пʼять та менше або дорівнює двом.

{

if (abs(x)<0.00000001) // Якщо модуль "х" дуже мале число.

{

continue; // Перехід в кінець тіла циклу.

}

else // Інакше.

{

y = 1/(pow(x,25)); // "у" приймає значення "х" у мінус двадцять пʼятому степені.

}

}

else // Інакше.

{

if (x > 2 && x < 4 ) // Якщо "х" строго більше двох і строго менше чотирьох.

{

y = pow( exp(2\*x) \* sqrt(x) - ((x + 1.0/3)/x) , 1.0/3 ) \* abs(cos(2.5\*x)); // "y" приймає значення кубічного кореня з числа "e" в ступені два помножине на "x" далі множиться на квадратний корінь із "x" і віднімається вираз, де в чисельнику "x" додати одну третю, а в знаменнику "x", і весь цей квадратний корінь множиться на модуль косинуса, аргументом якого є число два з половиною помножине на "x".

}

else // Інакше.

{

if (x<=-5) // Якщо "x" менше або дорівнює мінус пʼяти.

{

y = (-1) \* pow ( -x, 1.0/3 ); // "y" приймає значення мінус один помножине на мінус "х" у кубічному корені.

}

if (x>=4) // Якщо "х" більше або дорівнює чотирьом.

{

y = pow ( x, 1.0/3 ); // "y" приймає значення "х" у кубічному корені.

}

}

}

cout<<"X = "<< x << ", F(x) = " << y << "\n"; // Виводимо на консоль значення "x" і чому вони дорівнюють, і значення функції F(х) - "y", переходимо на новий рядок.

}

return 0; // Без повторів.

}

4.Опис інтерфейсу програми (керівництво користувача)

