

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 2

з дисципліни “Основи програмування”

тема “Математичні вирази та функції”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП-01  Беліцький Олександр Сергійович  (*прізвище, ім’я, по батькові*)  Варіант № 3 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

**Мета роботи**

Познайомитись із програмуванням на мові С#. Навчитись використовувати змінні числових типів даних. Навчитися на практиці проводити точні обчислення математичних формул за допомогою операторів та стандартних функцій. Навчитися виконувати компіляцію власного коду за допомогою утиліт С# .NET Core.

**Постановка завдання**

#### **Частина 1. Математичні формули**

Програмно виконати розрахунки заданих математичних формул:

,

де

,

Задати вхідні значення (a,b,c), вивести вхідні дані та результати обчислень () у консоль.

#### **Частина 2. Кусково-задана функція**

Дано функцію:

⎧ - 2 / (4 \* x - 1) - 1 , x є (-6, 6) U [8, 10]

y = ⎨

⎩ tg(x^2) + sin(2 \* x)^2 , else

Задати вхідне значення *x*, вивести вхідне значення та значення *y* у консоль.

**Аналіз вимог і проектування**

**Частина 1**

**Визначення ОДЗ:**

**1)**

не існує при *a = b*

**2)**

не існує при *a =* 0

**3)**

не існує при *с = -*1

**Частина 2**

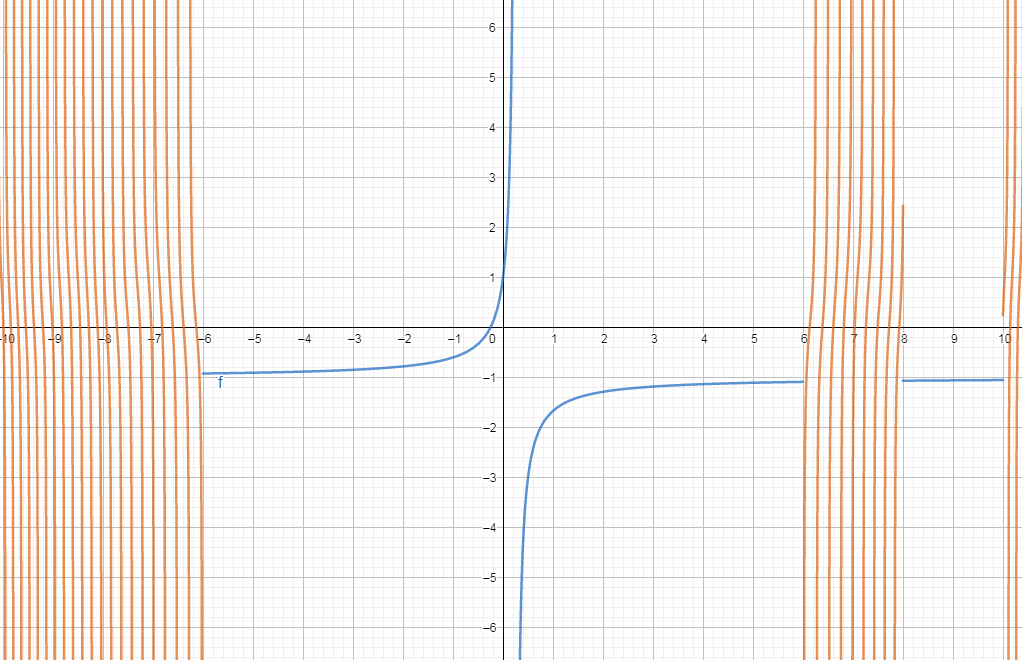
**Визначення ОДЗ:**

**1)**

*y* не існує при *х* = 0.25

**2)**

*y* не існує при або при



Графік *у(х)* на проміжку *х* ∈ [-10,10]

**Тексти коду програм**

**Частина 1**

|  |
| --- |
| **Program.cs** |
| using System;  using static System.Console;  using static System.Math;  namespace lab1  {  class Program  {  static void Main(string[] args)  {  WriteLine("Enter a: ");  double a = double.Parse(ReadLine());  WriteLine("Enter b: ");  double b = double.Parse(ReadLine());  WriteLine("Enter c: ");    double c = double.Parse(ReadLine());  double d0 = ((Pow((a+3), (c+1))-10)/(a-b));  double d1 = 4\*b+(c/a);  double d2 = Pow((a+4), ((Abs(Sin(b)))/(1+c)));  double d = d0+d1+d2;    WriteLine("d0 = {0}.", d0);  WriteLine("d1 = {0}.", d1);  WriteLine("d2 = {0}.", d2);  WriteLine("d = {0}.", d);  }  }  } |

**Частина 2**

|  |
| --- |
| **Program.cs** |
| using System;  using static System.Console;  using static System.Math;  namespace part2  {  class Program  {  static void Main(string[] args)  {  double n;  double y;  WriteLine("Enter x: ");  double x = double.Parse(ReadLine());  n = (Pow(x, 2) - PI / 2) / PI;  if ((x > -6 && x < 6) || (x >= 8 && x <= 10))  {  if (x != 0.25)  {  y = (-2 / (4 \* x - 1) - 1);  }  else  {  y = double.NaN;  }  }  else  {  if (n % Floor(n) != 0)  {  y = Tan(Pow(x, 2)) + Pow(Sin(2 \* x), 2);  }  else  {  y = double.NaN;  }  }  WriteLine("y = {0}.", y);  }  }  } |

**Приклади результатів**

**Частина 1**

При введенні чисел із ОДЗ:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab1/part1$ dotnet run Enter a:  0,2 Enter b:  -12,5 Enter c:  1,0 d0 = 0,01889763779527575. d1 = -45. d2 = 1,048739287586288. d = -43,932363074618436. |

При введенні чисел, що не входять до ОДЗ:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab1/part1$ dotnet run Enter a:  0,2 Enter b:  0,2 Enter c:  1 d0 = ∞. d1 = 5,8. d2 = 1,1532149390585784. d = ∞. |

**Частина 2**

При введенні *x* ∈ (-6, 6) U [8, 10] *x* ≠ 0.25:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab1/part2$ dotnet run  Enter x:  -5 y = -0,9047619047619048. |

При введенні *х* = 0.25:

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab1/part2$ dotnet run Enter x:  0,25 y = не число. |

При введенні *x* ∈ [-10, -6] U (6, 8) або :

|  |
| --- |
| alexandr:~/progbase/progbase/labs/lab1/part2$ dotnet run Enter x:  -8 y = 2,4307486289421814. |

**Висновки**

Виконавши дану лабораторну роботу були проведені точні обчислення математичних формул мови програмування C# за допомогою її операторів та функцій. Мова C# має засоби достатньо точної обробки числових типів даних.

Також, за допомогою конструкцій розгалуження, можливо виконувати різні обрахунки на основі вхідних даних та задавати у програмі умови обробки даних і прийняття рішень на основі них, що й було показано на практиці у Частині №2.

В результаті виконання Частини №2 було застосовано конструкції розгалуження для демонстрації спрощення програмного коду при прийнятті програмою рішень і обробки числових значень на основі вхідних даних.

Компіляція всього коду відбувалася за допомогою утиліти dotnet.