Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №5**

**По вычислительной математике**

**Вариант: 3**

Выполнил:

студент 2 курса

Батманов Даниил Евгеньевич

Группа: Р3207

Принял:

Рыбаков Степан Дмитриевич

Отчёт принят «\_\_»\_\_\_\_\_2024 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2024

**Оглавление**

[Задание 3](#_Toc162833322)

[Рабочие формулы 3](#_Toc162833323)

[Часть I: вычислительная 4](#_Toc162833324)

[Часть II: программная 5](#_Toc162833325)

[Вывод 13](#_Toc162833326)

# 

# 

# 

# Задание

**Вычислительная реализация задачи:**

1. Выбрать из табл. 1 заданную по варианту таблицу 𝑦 = 𝑓(𝑥) (таблица 1.1 –

таблица 1.5);

|  |  |  |  |
| --- | --- | --- | --- |
| **x** | **y** | **X1** | **X2** |
| 1,1 | 0,2234 | 1,121 | 1,482 |
| 1,25 | 1,2438 | 1,852 | 1,652 |
| 1,4 | 2,2644 | 1,168 | 1,463 |
| 1,55 | 3,2984 | 1,875 | 1,575 |
| 1,7 | 4,3222 | 1,189 | 1,491 |
| 1,85 | 5,3516 | 1,891 | 1,671 |
| 2 | 6,3867 | 1,217 | 1,473 |

2. Построить таблицу конечных разностей для заданной таблицы. Таблицу отра-

зить в отчете;

3. Вычислить значения функции для аргумента 𝑋1 (см. табл.1), используя

первую или вторую интерполяционную формулу Ньютона. Обратить внима-

ние какой конкретно формулой необходимо воспользоваться;

4. Вычислить значения функции для аргумента 𝑋2 (см. табл. 1), используя

первую или вторую интерполяционную формулу Гаусса. Обратить внимание

какой конкретно формулой необходимо воспользоваться;

***5. Подробные вычисления привести в отчете.***

**Программная реализация задачи:**

1. Исходные данные задаются тремя способами:

a) в виде набора данных (таблицы x,y), пользователь вводит значения с клавиа-

туры;

b) в виде сформированных в файле данных (подготовить не менее трех тестовых

вариантов);

c) на основе выбранной функции, из тех, которые предлагает программа, напри-

мер, sin(𝑥). Пользователь выбирает уравнение, исследуемый интервал и коли-

чество точек на интервале (не менее двух функций).

2. Сформировать и вывести таблицу конечных разностей;

3. Вычислить приближенное значение функции для заданного значения аргумен-

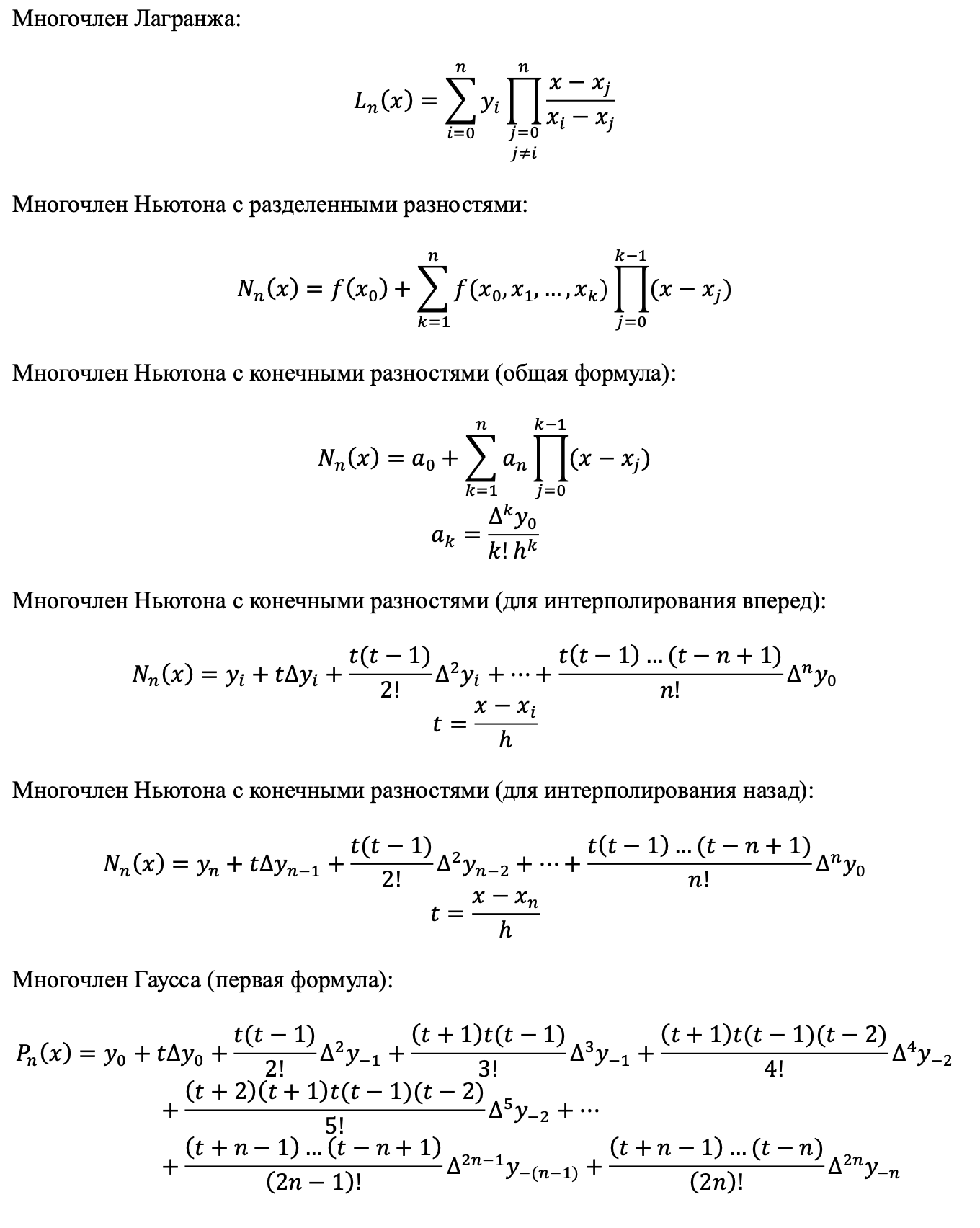
та, введенного с клавиатуры, указанными методами (см. табл. 2). Сравнить по-

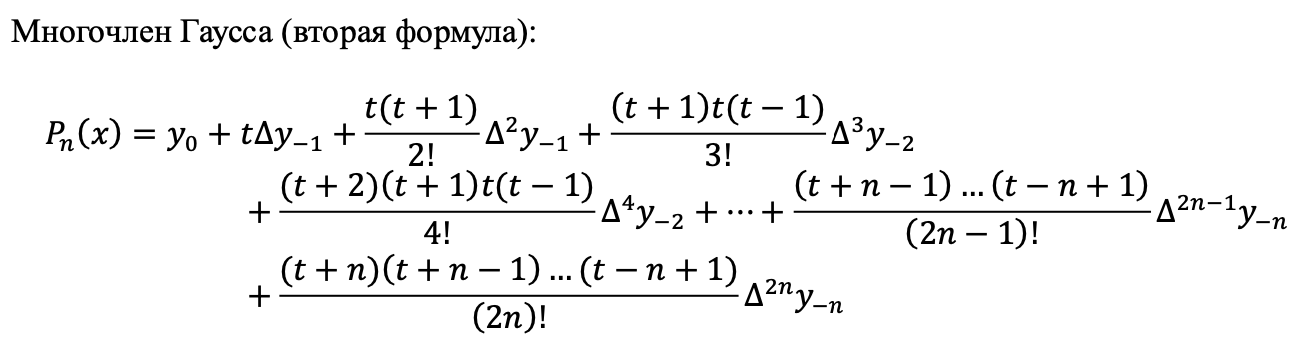
лученные значения;

4. Построить графики заданной функции с отмеченными узлами интерполяции и

интерполяционного многочлена Ньютона/Гаусса (разными цветами);

# Рабочие формулы





# Часть I: вычислительная

Таблица конечных разностей:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **x** | **y** | **dy** | **dy2** | **dy3** | **dy4** | **dy5** | **dy6** |
| **0** | 1,1 | 0,2234 | 1,0204 | 0,0002 | 0,0132 | -0,0368 | 0,0762 | -0,1313 |
| **1** | 1,25 | 1,2438 | 1,0206 | 0,0134 | -0,0236 | 0,0394 | -0,0551 |  |
| **2** | 1,4 | 2,2644 | 1,034 | -0,0102 | 0,0158 | -0,0157 |  |  |
| **3** | 1,55 | 3,2984 | 1,0238 | 0,0056 | 0,0001 |  |  |  |
| **4** | 1,7 | 4,3222 | 1,0294 | 0,0057 |  |  |  |  |
| **5** | 1,85 | 5,3516 | 1,0351 |  |  |  |  |  |
| **6** | 2 | 6,3867 |  |  |  |  |  |  |

Вычисление функции формулой Ньютона:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **y** | **X1** | **X2** | **t** | | **X1** | **N** |
| 1,1 | 0,2234 | 1,121 | 1,482 | **t1** | 0,14 | **x1** | 0,37147968 |
| 1,25 | 1,2438 | 1,852 | 1,652 | **t2** | 4,013333333 | **x2** | 4,33541831 |
| 1,4 | 2,2644 | 1,168 | 1,463 | **t3** | -1,546666667 | **x3** | 4,49635565 |
| 1,55 | 3,2984 | 1,875 | 1,575 | **t4** | 2,166666667 | **x4** | 2,4367259 |
| 1,7 | 4,3222 | 1,189 | 1,491 | **t5** | -3,406666667 | **x5** | 1,5742797 |
| 1,85 | 5,3516 | 1,891 | 1,671 | **t6** | 0,273333333 | **x6** | 0,50969456 |
| 2 | 6,3867 | 1,217 | 1,473 | **t7** | -5,22 | **x7** | 25,142563 |

=(C11-A11)/0,15

=C2+F11\*D2+(F11\*(F11-1))/2\*E2+(F11\*(F11-1)\*(F11-2))/6\*F2+(F11\*(F11-1)\*(F11-2)\*(F11-3))/24\*G2+(F11\*(F11-1)\*(F11-2)\*(F11-3)\*(F11-4))/120\*H2+(F11\*(F11-1)\*(F11-2)\*(F11-3)\*(F11-4)\*(F11-5))/(120\*6)\*I2

Вычисление функции формулой Гаусса:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **y** | **X1** | **X2** | **t** | | **X2** | **P** |
| 1,1 | 0,2234 | 1,121 | 1,482 | **t1** | 2,546666667 | **x1** | 5,92184066 |
| 1,25 | 1,2438 | 1,852 | 1,652 | **t2** | 2,68 | **x2** | 6,06021513 |
| 1,4 | 2,2644 | 1,168 | 1,463 | **t3** | 0,42 | **x3** | 3,72963968 |
| 1,55 | 3,2984 | 1,875 | 1,575 | **t4** | 0,166666667 | **x4** | 3,46983968 |
| 1,7 | 4,3222 | 1,189 | 1,491 | **t5** | -1,393333333 | **x5** | 1,85703028 |
| 1,85 | 5,3516 | 1,891 | 1,671 | **t6** | -1,193333333 | **x6** | 2,05761238 |
| 2 | 6,3867 | 1,217 | 1,473 | **t7** | -3,513333333 | **x7** | -0,5038903 |

=(D21-A21)/0,15

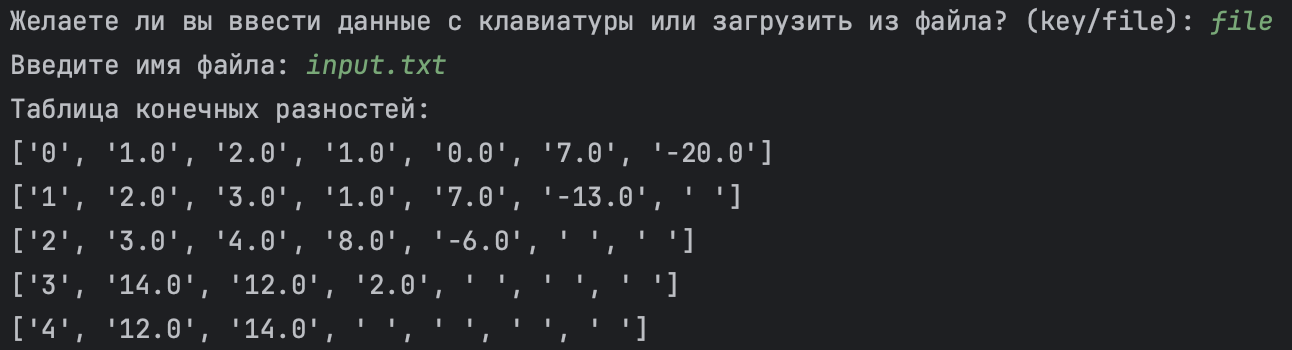
=$C$5+F21\*$D$5+(F21\*(F21-1))/2\*$E$4+((F21+1)\*F21\*(F21-1))/6\*$F$4+((F21+1)\*F21\*(F21-1)\*(F21-2))/24\*$G$3+((F21+2)\*(F21+1)\*F21\*(F21-1)\*(F21-2))/120\*$H$3+((F21+2)\*(F21+1)\*F21\*(F21-3)\*(F21-2)\*(F21-1))/(120\*6)\*$I$2

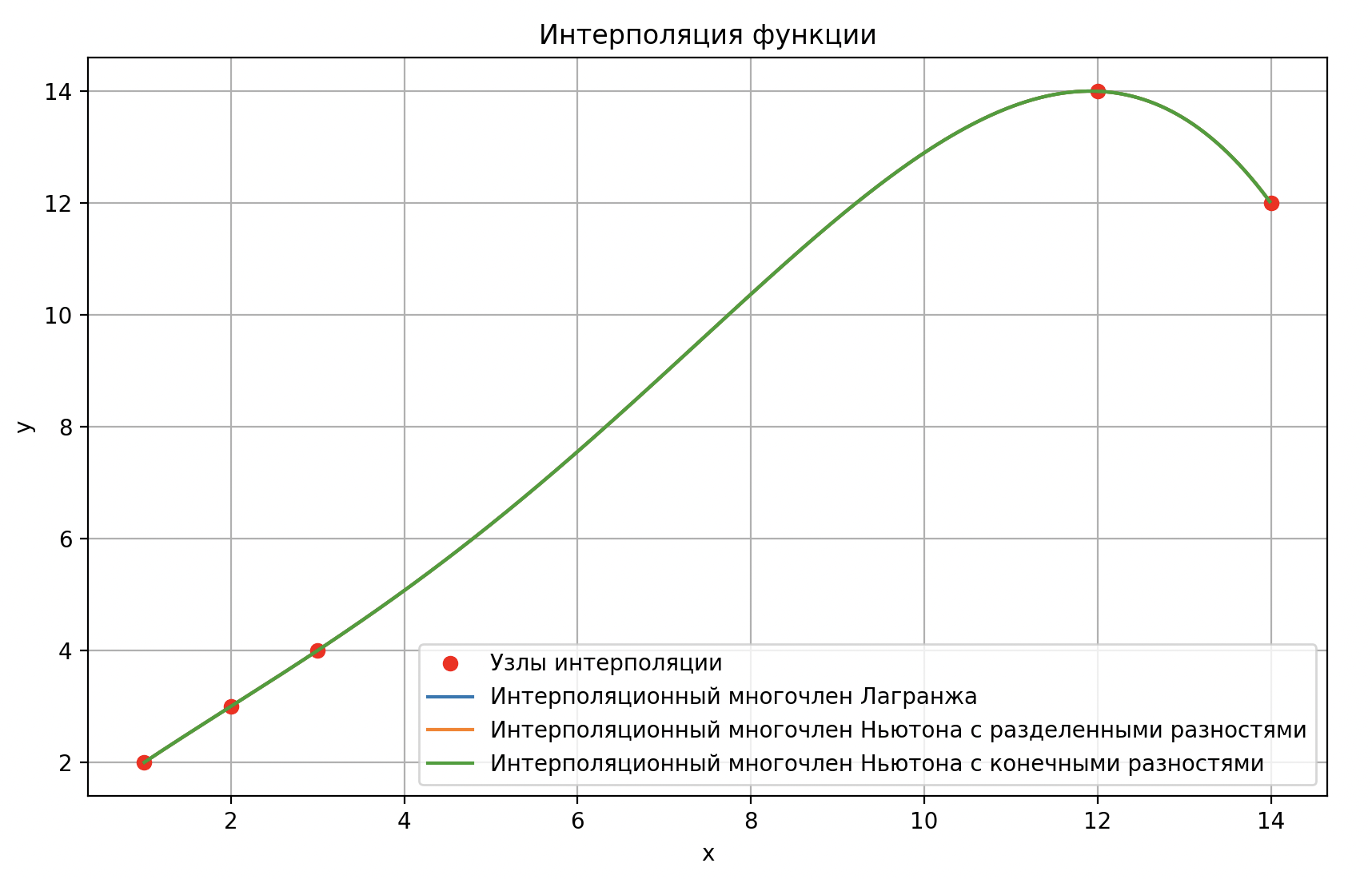
# Часть II: программная

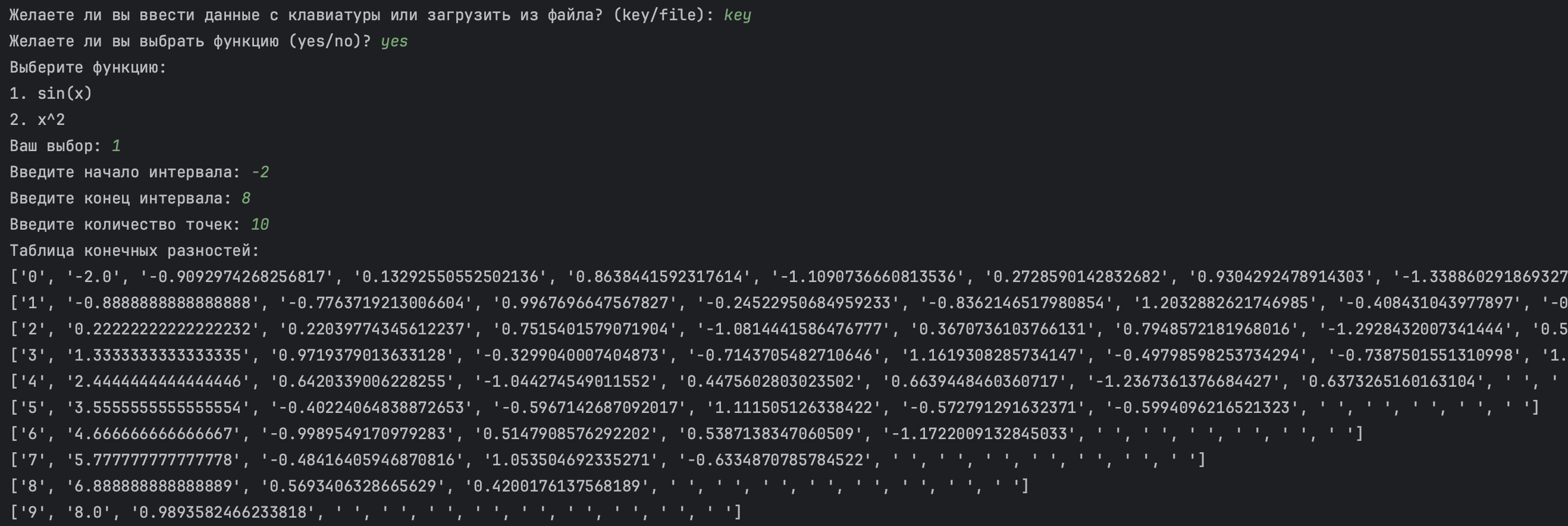
***Исходный код:***

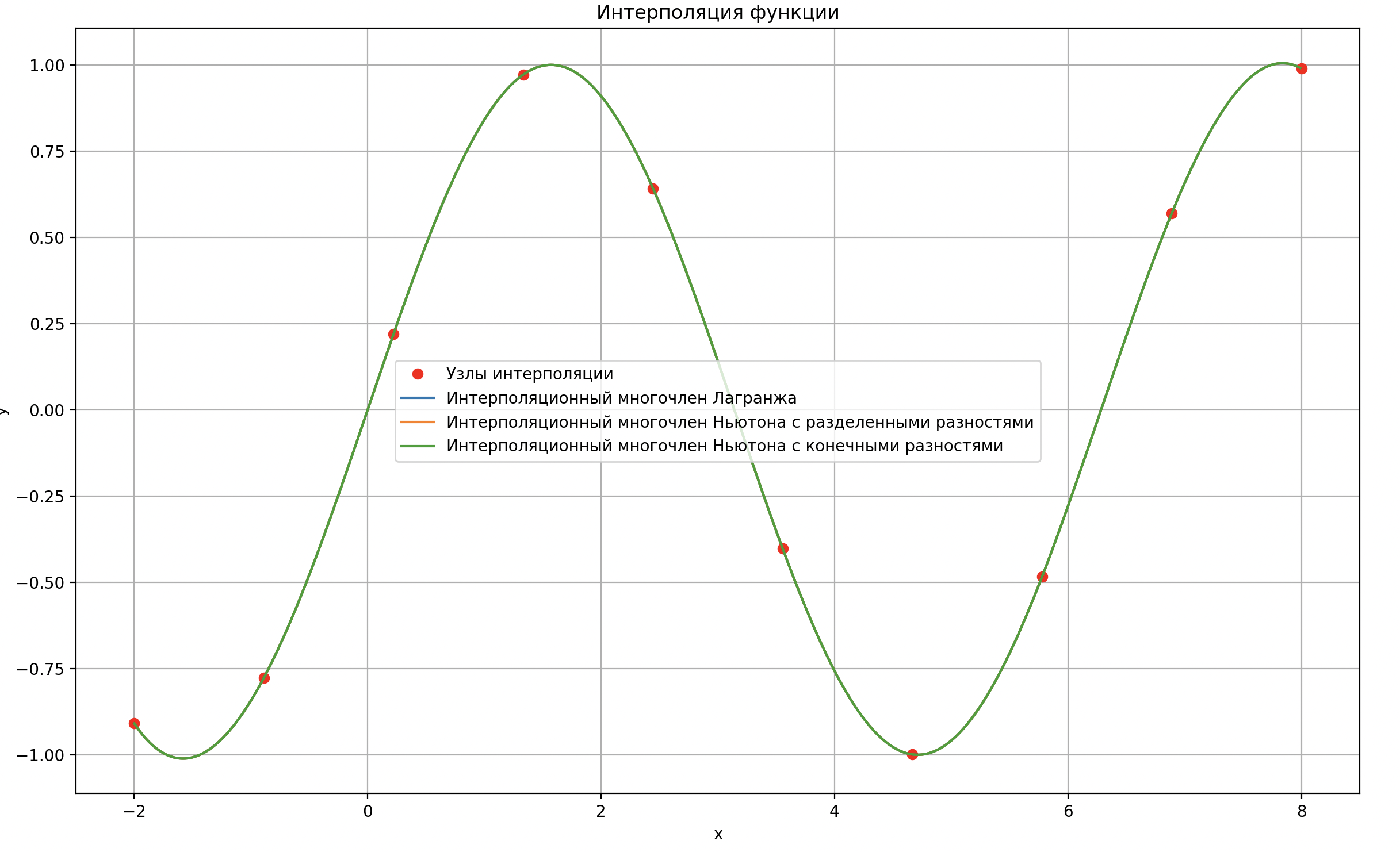
import math  
import matplotlib.pyplot as plt  
  
class method:  
 def \_\_init\_\_(self, x, y):  
 self.x = x  
 self.y = y  
 self.n = len(x)  
  
 def calculate(self, value):  
 pass  
  
class lagrange\_meth(method):  
 def calculate(self, value):  
 n = len(self.x)  
 l = [1] \* n  
  
 for i in range(n):  
 for j in range(n):  
 if i != j:  
 l[i] \*= (value - self.x[j]) / (self.x[i] - self.x[j])  
  
 res = 0  
 for i in range(n):  
 res += self.y[i] \* l[i]  
  
 return res  
  
  
class newton\_of\_divided\_differences\_meth(method):  
 def calculate(self, value):  
 res = 0  
  
 n = len(self.x)  
 for i in range(n):  
 part = 1  
  
 indexes = list(range(i + 1))  
 for j in range(i + 1):  
 if j != i:  
 part \*= value - self.x[j]  
  
 part \*= self.f(indexes)  
 res += part  
  
 return res  
  
 def f(self, indexes):  
 if len(indexes) == 1:  
 return self.y[indexes[0]]  
 return (self.f(indexes[1:]) - self.f(indexes[:-1])) / (self.x[indexes[-1]] - self.x[indexes[0]])  
  
  
class newton\_of\_finite\_differences\_meth(method):  
 def \_\_init\_\_(self, x, y, h):  
 super().\_\_init\_\_(x, y)  
 self.h = h  
 self.n = len(x)  
  
 def calculate(self, value):  
 res = 0  
  
 for k in range(self.n):  
 part = self.a(k)  
  
 for j in range(k):  
 part \*= value - self.x[j]  
  
 res += part  
  
 return res  
  
 def a(self, k):  
 return self.f(0, k) / (self.fact(k) \* self.h \*\* k)  
  
 def f(self, i, k):  
 if k == 0:  
 return self.y[i]  
 return self.f(i + 1, k - 1) - self.f(i, k - 1)  
  
 def fact(self, k):  
 if k == 0:  
 return 1  
 res = 1  
 for i in range(1, k + 1):  
 res \*= i  
 return res  
  
 def get\_finite\_differences\_table(self):  
 headers = ["№", "xi", "yi"]  
 for i in range(self.n - 1):  
 headers.append("∆" + (str(i + 1) if i > 0 else "") + "yi")  
  
 data = []  
 for i in range(self.n):  
 row = [str(i), str(self.x[i])]  
 for j in range(self.n):  
 if j < (self.n - i):  
 row.append(str(self.f(i, j)))  
 else:  
 row.append(" ")  
 data.append(row)  
  
 return data  
  
  
def input\_from\_keyboard():  
 n = int(input("Введите количество точек: "))  
 x = []  
 y = []  
 for i in range(n):  
 x.append(float(input(f"Введите x{i+1}: ")))  
 y.append(float(input(f"Введите y{i+1}: ")))  
 return x, y  
  
def load\_from\_file(filename):  
 x = []  
 y = []  
 with open(filename, 'r') as file:  
 for line in file:  
 data = line.strip().split(',')  
 x.append(float(data[0]))  
 y.append(float(data[1]))  
 return x, y  
  
def select\_function():  
 print("Выберите функцию:")  
 print("1. sin(x)")  
 print("2. x^2")  
 choice = int(input("Ваш выбор: "))  
 return choice  
  
def generate\_function\_data(func, start, end, points):  
 if func == 1:  
 return [start + (end - start) \* i / (points - 1) for i in range(points)], [math.sin(start + (end - start) \* i / (points - 1)) for i in range(points)]  
 elif func == 2:  
 return [start + (end - start) \* i / (points - 1) for i in range(points)], [(start + (end - start) \* i / (points - 1)) \*\* 2 for i in range(points)]  
 else:  
 print("Неправильный выбор функции.")  
 return None, None  
  
  
def main():  
 data\_source = input("Желаете ли вы ввести данные с клавиатуры или загрузить из файла? (key/file): ")  
  
  
 if data\_source.lower() == "key":  
 choice = input("Желаете ли вы выбрать функцию (yes/no)? ")  
 if choice.lower() == "yes":  
 function\_choice = select\_function()  
 if not function\_choice:  
 print("Некорректный выбор функции.")  
 return  
 start = float(input("Введите начало интервала: "))  
 end = float(input("Введите конец интервала: "))  
 points = int(input("Введите количество точек: "))  
 x, y = generate\_function\_data(function\_choice, start, end, points)  
 if not x or not y:  
 print("Не удалось сгенерировать данные.")  
 return  
 elif choice.lower() == "no":  
 x, y = input\_from\_keyboard()  
  
 if not x or not y:  
 print("Не удалось получить данные.")  
 return  
 start = min(x)  
 end = max(x)  
 points = len(x)  
 else:  
 print("Некорректный ввод. Пожалуйста, выберите 'да' или 'нет'.")  
 return  
 elif data\_source.lower() == "file":  
 filename = input("Введите имя файла: ")  
 x, y = load\_from\_file(filename)  
 if not x or not y:  
 print("Не удалось загрузить данные из файла.")  
 return  
 start = min(x)  
 end = max(x)  
 points = len(x)  
 else:  
 print("Некорректный ввод. Пожалуйста, выберите 'клавиатура' или 'файл'.")  
 return  
  
 lagrange = lagrange\_meth(x, y)  
 newton\_divided = newton\_of\_divided\_differences\_meth(x, y)  
 newton\_finite = newton\_of\_finite\_differences\_meth(x, y, 1)  
  
 print("Таблица конечных разностей:")  
 for row in newton\_finite.get\_finite\_differences\_table():  
 print(row)  
  
 plt.figure(figsize=(10, 6))  
 plt.plot(x, y, 'ro', label="Узлы интерполяции")  
 plt.xlabel('x')  
 plt.ylabel('y')  
 plt.title('Интерполяция функции')  
 plt.grid(True)  
  
 x\_interpolated = [start + (end - start) \* i / 1000 for i in range(1000)]  
 y\_lagrange = [lagrange.calculate(val) for val in x\_interpolated]  
 y\_newton\_divided = [newton\_divided.calculate(val) for val in x\_interpolated]  
 y\_newton\_finite = [newton\_divided.calculate(val) for val in x\_interpolated]  
  
 plt.plot(x\_interpolated, y\_lagrange, label="Интерполяционный многочлен Лагранжа")  
 plt.plot(x\_interpolated, y\_newton\_divided, label="Интерполяционный многочлен Ньютона с разделенными разностями")  
 plt.plot(x\_interpolated, y\_newton\_finite, label="Интерполяционный многочлен Ньютона с конечными разностями")  
  
 plt.legend()  
 plt.show()  
  
 print("░░░░░░░░░░░░░░░░░░▄▄░░░░░░░░░░░░░░░░░░\n"  
 "░░░░░░░░░░░░░░░░░▄██▄░░░░░░░░░░░░░░░░░\n"  
 "░░░░░░░░░░░░░░░░▄████▄░░░░░░░░░░░░░░░░\n"  
 "░░░░░░░░░░░░░░░░██████░░░░░░░░░░░░░░░░\n"  
 "███▄▄░░░░░░░░░░████████░░░░░░░░░░▄▄▄██\n"  
 "▀██████▄▄░░░░░░████████░░░░░░▄▄██████▀\n"  
 "░█████████▄░░░░████████░░░░▄█████████░\n"  
 "░░▀▀▀▀▀▀▀▀▀▀░░░▀▀▀▀▀▀▀▀░░░▀▀▀▀▀▀▀▀▀▀░░\n"  
 "░░░▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄░░░\n"  
 "░░░░▀████████████████████████████▀░░░░\n"  
 "░░░░░░▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄░░░░░░\n"  
 "░░░░░░░▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀░░░░░░░\n"  
 "░░░░░░░░░░░▄▄▄▄▄▄░▄▄░▄▄▄▄▄▄░░░░░░░░░░░\n"  
 "░░░░░░░░░░░░░░▀▀▀░▀▀░▀▀▀░░░░░░░░░░░░░░\n"  
 "░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░\n"  
 "░░░░░░░░░░░░██░▀▀░░░░░░██░░░░░░░░░░░░░\n"  
 "░▄█▀▀██░▄█▀▀██░██░▄█▀▀███░▄█▀▀██░█▀▀█▄\n"  
 "░█░░░██░█░░░██░██░█░░░███░█░░░██░▀███▄\n"  
 "░▀█▄▄██░▀█▄▄██░██░▀█▄▄███░▀█▄▄██░█▄▄█▀")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

***Вывод программы:***

******

******

******

******

# Вывод

В ходе выполнения данной лабораторной работы я решил задачу интерполяции, нашёл значения функции при заданных значениях аргумента, отличных от узловых точек.

░░░░░░░░░░░░░░░░░░▄▄░░░░░░░░░░░░░░░░░░

░░░░░░░░░░░░░░░░░▄██▄░░░░░░░░░░░░░░░░░

░░░░░░░░░░░░░░░░▄████▄░░░░░░░░░░░░░░░░

░░░░░░░░░░░░░░░░██████░░░░░░░░░░░░░░░░

███▄▄░░░░░░░░░░████████░░░░░░░░░░▄▄▄██

▀██████▄▄░░░░░░████████░░░░░░▄▄██████▀

░█████████▄░░░░████████░░░░▄█████████░

░░▀▀▀▀▀▀▀▀▀▀░░░▀▀▀▀▀▀▀▀░░░▀▀▀▀▀▀▀▀▀▀░░

░░░▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄░░░

░░░░▀████████████████████████████▀░░░░

░░░░░░▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄░░░░░░

░░░░░░░▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀░░░░░░░

░░░░░░░░░░░▄▄▄▄▄▄░▄▄░▄▄▄▄▄▄░░░░░░░░░░░

░░░░░░░░░░░░░░▀▀▀░▀▀░▀▀▀░░░░░░░░░░░░░░

░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░

░░░░░░░░░░░░██░▀▀░░░░░░██░░░░░░░░░░░░░

░▄█▀▀██░▄█▀▀██░██░▄█▀▀███░▄█▀▀██░█▀▀█▄

░█░░░██░█░░░██░██░█░░░███░█░░░██░▀███▄

░▀█▄▄██░▀█▄▄██░██░▀█▄▄███░▀█▄▄██░█▄▄█▀

ЧЁТКО