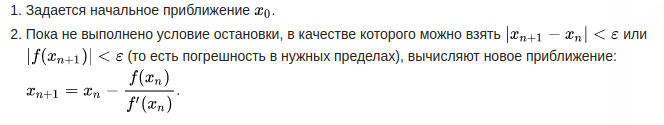
Лабораторная работа №6

**Тема:** Нахождение корней нелинейных уравнений (методом касательных, хорд, деления отрезка пополам)

**Оборудование:** VS Сode, python3

**Задача 1**

**Постановка задачи:**Найти корень уравнения на отрезке [1; 2] с точностью . Индивидуальное задание: найти корень уравнения на отрезке [0; 10] с точностью .

**Математическая модель:** решение уравнения методом Ньютона (касательных): 

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Предназначение переменной | Тип переменной |
| a,b | Границы поиска корня | float |
| cntA,cntB | Счетчик цикла | float |
| x,xi | Результат вычислений | float |
| e | Точность вычислений | float |

**Код программы:**

import math

def f(arg):

return arg\*\*3 - 2 \* arg\*\*2 + 3 \* arg - 5

def df(arg):

return 3 \* arg\*\*2 - 4 \* arg + 3

def ddf(arg):

return 6 \* arg - 4

def newton():

a = 1

b = 2

e = 0.000001

cntA = 0

cntB = 0

x = a

xi = x-f(x)/df(x)

cntA += 1

while abs(xi - x) > e:

x = xi

xi = x - f(x)/df(x)

cntA += 1

print("Answer A", round(xi, 6),"Count of cycles", cntA)

x = b

xi = x - f(x)/df(x)

cntB += 1

while abs(xi - x) > e:

x = xi

xi = x - f(x)/df(x)

cntB += 1

print("Answer B", round(xi, 6),"Count of cycles", cntB)

def f\_ind(arg):

return arg \* math.log10(arg+1) - 1

def df\_ind(arg):

return (arg+(arg+1)\*math.log(arg+1))/((arg+1)\*math.log(10))

def ddf\_ind(arg):

return (arg+2)/((arg+1)\*(arg+1)\*math.log(10))

def ind\_newton():

a = 0

b = 10

e = 0.000001

cntB = 0

x = b

xi = x - f\_ind(x)/df\_ind(x)

cntB += 1

while abs(xi - x) > e:

x = xi

xi = x - f\_ind(x)/df\_ind(x)

cntB += 1

print("Answer B", round(xi, 6),"Count of cycles", cntB)

def main():

print("x^3-2\*x^2+3\*x-5=0")

newton()

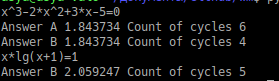
print("x\*lg(x+1)=1")

ind\_newton()

if \_\_name\_\_ == "\_\_main\_\_":

main()

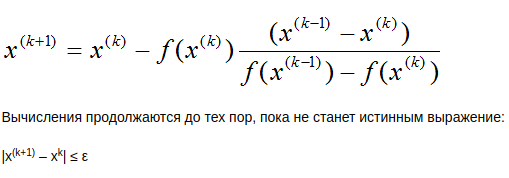
**Результат работы программы:**

****

**Задача 2**

**Постановка задачи:** Найти корень уравнения на отрезке [-3; 4] с точностью . Индивидуальное задание: найти корень уравнения на отрезке [0; 10] с точностью .

**Математическая модель:** решение уравнения методом хорд:



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Предназначение переменной | Тип переменной |
| a,b | Границы поиска корня | float |
| k | Счетчик цикла | float |
| x,xi | Результат вычислений | float |
| c | Промежуточная переменная | float |
| x0 | Начальное значение x | float |
| eps | Точность вычислений | float |

**Код программы:**

import math

from prettytable import PrettyTable

def f(arg):

return arg\*\*3-12\*arg-8

def find\_chord():

a = -3

b = 4

eps = 0.0001

x = a

if (3\*x\*\*2-12)\*(6\*x) > 0:

x0 = a

c = b

else:

x0 = b

c = a

print("a =", a, "b =", b, "eps =", eps, "x0 =", x0)

t = PrettyTable()

t.field\_names = ["x", "xi", "k"]

k = 0

xi = 0

while abs(xi - x) > eps:

x = x0

xi = x - (f(x)\*(c - x))/(f(c)-f(x))

x0 = xi

t.add\_row([round(x, 4), round(xi, 4), k])

k += 1

print(t.get\_string(title="f(x)"))

return xi

def f\_ind(arg):

return arg \* math.log10(arg+1) - 1

def find\_chord\_ind():

a = 0

b = 10

eps = 0.000001

x = b

if (x+(x+1)\*math.log(x+1))/((x+1)\*math.log(10)) \* ( (x+2)/( (x+1)\*(x+1)\*math.log(10) ) ) > 0:

x0 = a

c = b

else:

x0 = b

c = a

print("a =", a, "b =", b, "eps =", eps, "x0 =", x0)

t = PrettyTable()

t.field\_names = ["x", "xi", "k"]

k = 0

xi = 0

while abs(xi - x) > eps:

x = x0

xi = x - (f\_ind(x)\*(c - x))/(f\_ind(c)-f\_ind(x))

x0 = xi

t.add\_row([round(x, 6), round(xi, 6), k])

k += 1

print(t.get\_string(title="f(x) ind"))

return xi

def main():

a = find\_chord()

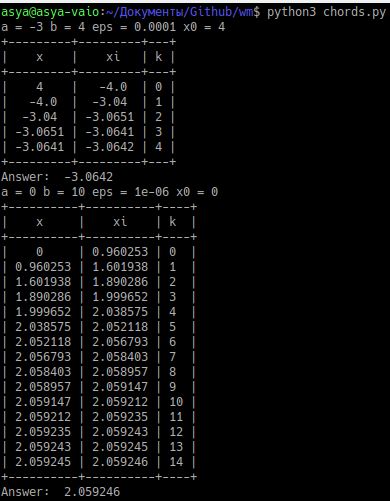
print("Answer: ", round(a, 4))

b = find\_chord\_ind()

print("Answer: ", round(b, 6))

if \_\_name\_\_ == "\_\_main\_\_":

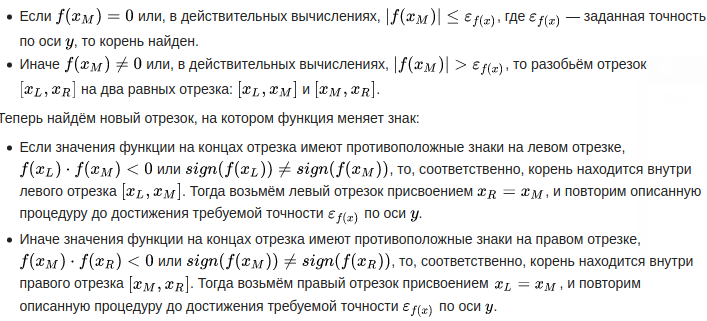
main()

**Результат работы программы:**

**Задача**

**Постановка задачи:**Индивидуальное задание: найти корень уравнения на отрезке [0; 10] с точностью .

**Математическая модель:** решение уравнения методом деления отрезка пополам:

****

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| Имя переменной | Предназначение переменной | Тип переменной |
| a,b | Границы поиска корня | float |
| x,xi | Результат вычислений | float |
| eps | Точность вычислений | float |
| xd | Середина отрезка | float |

**Код программы:**

import math

from prettytable import PrettyTable

def f\_ind(arg):

return arg \* math.log10(arg+1) - 1

def bisec():

a = 0

b = 10

eps = 0.000001

print("a =", a, "b =", b, "eps =", eps)

t = PrettyTable()

t.field\_names = ["x", "y"]

while (b-a) > eps:

xd = (b - a)/2

xi = a+xd

if f\_ind(a)\*f\_ind(xi) < 0:

b = xi

else:

a = xi

t.add\_row([round(xi, 6), round(f\_ind(xi), 6)])

t.align = "l"

print(t.get\_string(title="f(x) ind"))

return xi

def main():

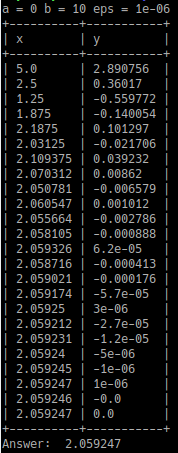
a = bisec()

print("Answer: ", round(a, 6))

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Результат работы программы:**

****