NC Lab 5
11762 Muhammad Kashif
Mullar Method

Task 1 x**3-2*x**2-5

Code

```
import sympy as s
import math as m
import cmath
def h1(x0,x2):
    return x0-x2
def h2(x1,x2):
    return x1-x2
def d1(x0,x2):
    return f(x0)-f(x2)
def d2(x1,x2):
    return f(x1)-f(x2)
def f(x):
    return x**3-2*x**2-5
def A(x0,x1,x2):
    a = (h1(x0,x2)*d2(x1,x2))-(h2(x1,x2)*d1(x0,x2))
    b =h1(x0,x2)*h2(x1,x2)*(x1-x0)
    return a/b
def B(x0,x1,x2):
    a = ((h1(x0,x2))**2*d2(x1,x2))-((h2(x1,x2))**2*d1(x0,x2))
    b = h1(x0,x2)*h2(x1,x2)*(x0-x1)
    return a/b

def Muller(x0,x1,x2):
    for i in range(25):
        if B(x0,x1,x2)<0:
            Xr=x0-((2*f(x2))/(B(x0,x1,x2)-m.sqrt(B(x0,x1,x2)**2-(4*A(x0,x1,x2)*f(x1)))))
        else:
            Xr=x0-((2*f(x2))/(B(x0,x1,x2)+m.sqrt(B(x0,x1,x2)**2-(4*A(x0,x1,x2)*f(x1)))))
        print(i,"Xr ",Xr)
        tol = abs(Xr - x2)
        if tol < 0.01:
            break
        x0 = x1
        x1=x2
        x2=Xr

Muller(1.9,2,2.1)
```

Output

```
0 Xr  2.509264135110312
1 Xr  2.1688437773525604
2 Xr  2.724694183504837
3 Xr  2.480028239825771
4 Xr  2.4136677187599034
5 Xr  3.0102456053938536
6 Xr  2.2170158739548804
```

Task 2 x**3+(4.001*x**2)+1.101

Code

```
import sympy as s
import math as m

def h1(x0,x2):
    return x0-x2
def h2(x1,x2):
    return x1-x2
def d1(x0,x2):
    return f(x0)-f(x2)
def d2(x1,x2):
    return f(x1)-f(x2)
def f(x):
    return x**3+(4.001*x**2)+1.101
def A(x0,x1,x2):
    a = (h1(x0,x2)*d2(x1,x2))-(h2(x1,x2)*d1(x0,x2))
    b =h1(x0,x2)*h2(x1,x2)*(x1-x0)
    return a/b
def B(x0,x1,x2):
    a = ((h1(x0,x2))**2*d2(x1,x2))-((h2(x1,x2))**2*d1(x0,x2))
    b = h1(x0,x2)*h2(x1,x2)*(x0-x1)
    return a/b

def Muller(x0,x1,x2):
    for i in range(25):
        if B(x0,x1,x2)<0:
            Xr=x0-((2*f(x2))/(B(x0,x1,x2)-m.sqrt(B(x0,x1,x2)**2-(4*A(x0,x1,x2)*f(x1)))))
        else:
            Xr=x0-((2*f(x2))/(B(x0,x1,x2)+m.sqrt(B(x0,x1,x2)**2-(4*A(x0,x1,x2)*f(x1)))))
        print(i,"Xr ",Xr)
        tol = f(Xr)
        if tol < 0.01:
            break
        if Xr < x1:
            x2=x1
            x1 = Xr
        else:
            x0 = x1
            x1 = Xr

Muller(2.5,3,3.5)
```

Output

```
0 Xr  0.4884100288837509 6
1 Xr  1.1998635849147987
2 Xr  -0.9419871024465378
3 Xr  -0.3139258719556375
4 Xr  -1.7917978861136177
```

task 3 (10*x**3)-(8.3*x**2)+(2.295*x)-0.21141

Code

```
import sympy as s
import math as m

def h1(x0,x2):
    return x0-x2
def h2(x1,x2):
    return x1-x2
def d1(x0,x2):
    return f(x0)-f(x2)
def d2(x1,x2):
    return f(x1)-f(x2)
def f(x):
    return (10*x**3)-(8.3*x**2)+(2.295*x)-0.21141
def A(x0,x1,x2):
    a = (h1(x0,x2)*d2(x1,x2))-(h2(x1,x2)*d1(x0,x2))
    b =h1(x0,x2)*h2(x1,x2)*(x1-x0)
    return a/b
def B(x0,x1,x2):
    a = ((h1(x0,x2))**2*d2(x1,x2))-((h2(x1,x2))**2*d1(x0,x2))
    b = h1(x0,x2)*h2(x1,x2)*(x0-x1)
    return a/b

def Muller(x0,x1,x2):
    for i in range(25):
        if B(x0,x1,x2)<0:
            Xr=x0-((2*f(x2))/(B(x0,x1,x2)-m.sqrt(B(x0,x1,x2)**2-(4*A(x0,x1,x2)*f(x1)))))
        else:
            Xr=x0-((2*f(x2))/(B(x0,x1,x2)+m.sqrt(B(x0,x1,x2)**2-(4*A(x0,x1,x2)*f(x1)))))
        print(i,"Xr ",Xr)
        tol = f(Xr)
        if tol < 0.0001:
            break
        if Xr < x1:
            x2=x1
            x1 = Xr
        else:
            x0 = x1
            x1 = Xr


Muller(3.5,4,4.5)
```

Output

```
0 Xr  1.5033164301044897
1 Xr  2.208406776594141
2 Xr  0.05860102076168139
```