NEWTON FORWARD INTERPOLATION

**n = int(input("enter total number of elements"))**

**x,y=[],[]**

**for i in range(0,n):**

**x.append(float(input("enter elements of x: ")))**

**y.append(float(input("enter elements of y :")))**

**b = []**

**b.append(y)**

**for i in range(0,n-1):**

**a = []**

**for j in range(0,n-1):**

**a.append(y[j+1]-y[j])**

**b.append(a)**

**n-=1**

**y=a**

**print(b)**

**print("==========formula-part============")**

**m = len(x)**

**value = float(input("y(x) : enter x: "))**

**p = (value- x[0])/(x[1] - x[0])**

**Temp = []**

**import math**

**temp = 1**

**for i in range(0,m-1):**

**temp = temp \*(p-i)**

**Temp.append(temp)**

**umair = b[0][0]**

**for i in range(0,m-1):**

**umair = umair + ((Temp[i]\*b[i+1][0])/(math.factorial(i+1)))**

**print(f"\nvalue at f({value}) is , {round(umair,5)}")**

**OUTPUT:**

**enter total number of elements5**

**enter elements of x: 1**

**enter elements of y :1**

**enter elements of x: 3**

**enter elements of y :27**

**enter elements of x: 5**

**enter elements of y :125**

**enter elements of x: 7**

**enter elements of y :343**

**enter elements of x: 9**

**enter elements of y :729**

**[[1.0, 27.0, 125.0, 343.0, 729.0], [26.0, 98.0, 218.0, 386.0], [72.0, 120.0, 168.0], [48.0, 48.0], [0.0]]**

**==========formula-part============**

**y(x) : enter x: 2**

**value at f(2.0) is , 8.0**