PAF KIET	**COLLEGE OF COMPUTING AND INFORMATION SCIENCES** **Final Assessment of Lab Exam (Fall 2020 Semester)**

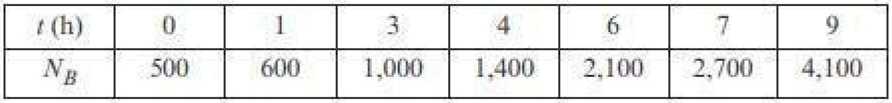
Class Id	105127	**Course Title**	Numerical Computing
Program	BSCS	**Campus / Shift**	North/ Morning
Date	November 26, 2020	**Total Marks**	20
Duration	03 hours	**Faculty Name**	Iqra Fahad
Student Id	63961	**Student Name**	Ali Afzal
Code	**A**		

Instructions:

- Fill out your Student ID and Student Name in above header.
- Do not remove or change any part question paper.
- Write down your answers with title “Answer for Question# 00”.
- Handwritten text or image should be on A4 size page with clear visibility of contents.
- In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
- Viva can be taken with prior notice, where deemed necessary.
- **Caution:** Duration to perform Final Assessment is **02 hours only and 01 hour** is given to cater all kinds of odds in submission of Answer-sheet. **Therefore, if you failed to upload answer sheet on LMS (in PDF format) within 3 hours’ limit, you would be considered as ABSENT/FAILED.**

Question 01:

The number of bacteria measured at different times t is given in the following table. Determine a function that best fits the data (built-in function allowed). Use the equation to estimate the number of bacteria after 5h (built-in function not allowed). Make a plot of the points and the equation.

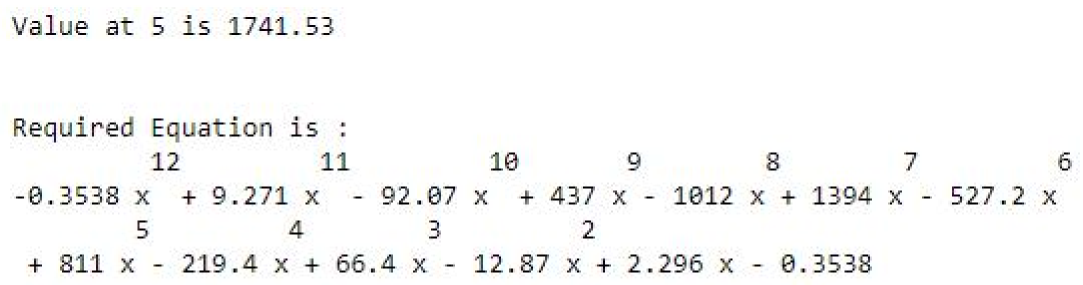


t (h)	0	1	3	4	6	7	9
N_B	500	600	1,000	1,400	2,100	2,700	4,100

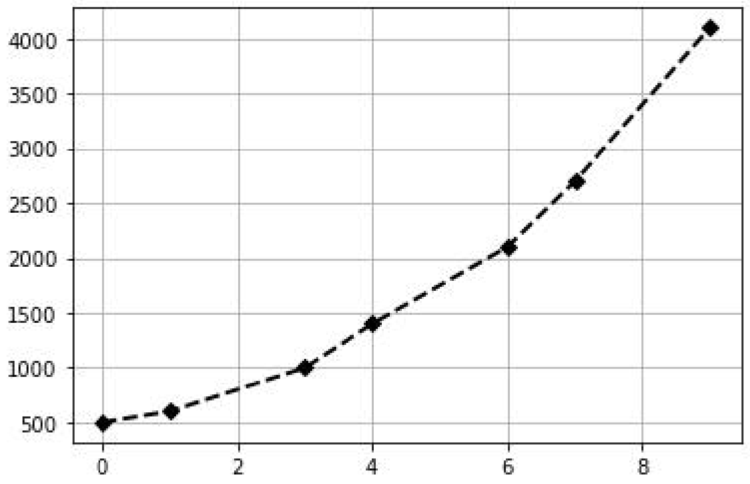
Answer 01:

```
import matplotlib.pyplot as plt
from scipy.interpolate import lagrange


def proterm(i, value, x):
    pro = 1
    for j in range(i):
        pro = pro * (value - x[j])
    return pro



def dividedDiffTable(x, y, n):
    for i in range(1, n):
        for j in range(n - i):
            y[j][i] = ((y[j][i - 1] - y[j + 1][i - 1])/(x[j] - x[i + j]))
    return y



def applyFormula(value, x, y, n):
    sum = y[0][0]
    for i in range(1, n):
        sum = sum + (proterm(i, value, x) * y[0][i])
    return sum



n = 7


y = [[0 for i in range(7)] for j in range(7)]


x = [ 0, 1, 3, 4, 6, 7, 9]
```

```
y[0][0] = 500
y[1][0] = 600
y[2][0] = 1000
y[3][0] = 1400
y[4][0] = 2100
y[5][0] = 2700
y[6][0] = 4100



y=dividedDiffTable(x, y, n)


value = 5


print("\nValue at", value, "is", round(applyFormula(value, x, y, n), 2))
print("\n\nRequired Equation is :")
print(lagrange(x,y))
print("\n\n")


ploty=[]
for i in range(7):
    ploty.append(y[i][0])


plt.plot(x,ploty,color='k', marker='D',linestyle='--', linewidth=2)
plt.grid()
plt.show()
```

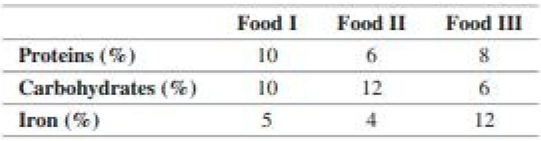


```
Value at 5 is 1741.53


Required Equation is :
         12         11         10       9        8        7         6
-0.3538 x  + 9.271 x  - 92.07 x  + 437 x - 1012 x + 1394 x - 527.2 x
        5         4        3         2
 + 811 x - 219.4 x + 66.4 x - 12.87 x + 2.296 x - 0.3538
```



Question 02

A dietitian wishes to plan a meal around three foods. The percentage of the daily requirements of proteins, carbohydrates, and iron contained in each ounce of the three foods is summarized in the following table:



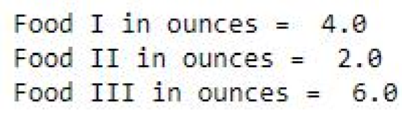
	Food I	Food II	Food III
Proteins (%)	10	6	8
Carbohydrates (%)	10	12	6
Iron (%)	5	4	12

Determine (using Python code) how many ounces of each food the dietitian should include in the meal to meet exactly the daily requirement of proteins, carbohydrates, and iron (100% of each).

Answer 02:

```
import numpy as np


n = 3
a = np.zeros((3,4))
x = np.zeros(3)


a[0][0]=10
a[0][1]=6
a[0][2]=8
a[0][3]=100
a[1][0]=10
a[1][1]=12
a[1][2]=6
a[1][3]=100
a[2][0]=5
a[2][1]=4
a[2][2]=12
a[2][3]=100


for i in range(n):
    for j in range(i+1, n):
        ratio = a[j][i]/a[i][i]
        for k in range(n+1):
            a[j][k] = a[j][k] - ratio * a[i][k]
```

```
x[n-1] = a[n-1][n]/a[n-1][n-1]
for i in range(n-2,-1,-1):
    x[i] = a[i][n]
    for j in range(i+1,n):
        x[i] = x[i] - a[i][j]*x[j]
    x[i] = x[i]/a[i][i]


print("Food I in ounces = ",x[0])
print("Food II in ounces = ",x[1])
print("Food III in ounces = ",x[2])
```



```
Food I in ounces =  4.0
Food II in ounces =  2.0
Food III in ounces =  6.0
```