

Numeric Analysis - Final Project

QuestionLottery

GitHub: <https://github.com/Maor-Ar/Numerical-Analysis-Hackathon/blob/main/Q1QuestionLottery.py>

חברי הקבוצה

שם הסטודנט	ת.ז.
נריה זודי	207073545
מתן אוחיון	311435614
מאור ארנון	205974553
מתן סופר	208491811

קובץ הגרלת שאלות:

הגרלת השאלות שלנו התבצעה ע"י שימוש באינטרפולציה לגראנז' ע"פ תעודות הזהות של חברי הקבוצה בחרנו מספרים שהם נקודות הטבלה לפי ספרות של מספרי תעודות הזהות. הספרות האחרונות של תעודות הזהות יהיו הערכים שעליהם נבצע אינטרפולציה את התוצאות עיגלנו כלפי מטה (כדי להימנע ממספרים לא שלמים) לאחר מכן עשינו פעולת מודולו כדי לבחור שאלה אחת (או שתיים) מטווח השאלות בחלק הנתון

השאלות שנבחרו:

חלק 1 – שאלה 4
חלק 2 – שאלות 12,14
חלק 3 – שאלות 28,30
חלק 4 – שאלה 32

להלן הסבר על האלגוריתם וקוד הממשש את הבחירה:

האלגוריתם מחזיר שאלות קבועות

הזרח שאלות

תעודות להיות:

$$\{3, 1, 5, 4\} V$$

(2.2, 2.3)	P_0
(0.0, 0.1)	P_1
(5.8, 7.1)	P_2
(9.4, 0.4)	P_3
(7.9, 7.3)	P_4
(4.1, 3.5)	P_5
(5.8, 5.6)	P_6
(5.1, 4.1)	P_7



2	0	5	9	7	4	5	5	3	מאור. א -
2	0	8	4	9	1	8	1	1	מית. ס -
2	0	7	0	7	3	5	4	5	נריה. ב -
3	1	1	4	3	5	6	1	4	מתן. א -

שאלה / קט	נקודות	דרך אינטר פולציה
שאלה 1	P_0, P_1	$V_0 \cdot V_1$
שאלה 2, 3	P_2, P_3	V_0, V_1
שאלה 4, 5	P_4, P_5	V_2, V_3
שאלה 6	P_6, P_7	$V_2 \cdot V_3$

כל ערך שנקבע נחס שאלה צי עיכול הערך כלפי מטה
ומוצאו מתאים למספר השאלות באותו החלק

עדימנה נציג לחלק 4 יצא 171.5

$$\text{Output} \rightarrow 171.5 \xrightarrow{\text{floor}} 171 \xrightarrow{\frac{5+1}{(5-1)}} 2 \xrightarrow{\frac{31}{(4-5)}} 33$$

(5 שאלות)
(4-5)
(4 שאלות)

פונקציה לחישוב אינטרפולציה לפי שיטת לגראנז

```
def Lagrange_interpolation(Table,pointToFindVal): ##calculate by Lagrange
    yp=0

    for i in range(len(Table)):
        p=1
        for j in range(len(Table)):
            if i!=j:
                p = p * (pointToFindVal - Table[j][0])/(Table[i][0]-Table[j][0])
        yp= yp + p * Table[i][1]

    print("For points -" ,Table , "And x value-" ,pointToFindVal , "We get -" , yp)
    return yp
```

פונקציה לבחירת שאלות לפי מספרי תעודות הזהות של חברי הקבוצה

```
def HackathonQuestionLottery():
    table_Points = [[2.2,2.3],[0.0,0.1],[5.8,7.1],[9.4,0.4],[7.9,7.3],[4.1,3.5],[5.8,5.6],[5.1,4.1]]
    values = [3,1,5,4]

    a = Lagrange_interpolation(table_Points[0:2],values[0]*values[1])
    b,c = Lagrange_interpolation(table_Points[0:2],values[0]), Lagrange_interpolation(table_Points[0:2],values[1])
    d,e = Lagrange_interpolation(table_Points[2:4],values[2]), Lagrange_interpolation(table_Points[2:4],values[3])
    f = Lagrange_interpolation(table_Points[6:8],values[2]*values[3])
    print("Returned Values ->" ,a,b,c,d,e,f)
    print(bcolors.OKBLUE, "from Part 1 do: ", math.floor(a) % 9 + 1, bcolors.ENDC)
    print(bcolors.OKBLUE, "from Part 2 do: ", math.floor(b) % 8 + 11, "And " ,math.floor(c) % 8 + 11, bcolors.ENDC)
    print(bcolors.OKBLUE, "from Part 3 do: ", math.floor(d) % 11 + 20, "And " ,math.floor(e) % 11 + 20, bcolors.ENDC)
    print(bcolors.OKBLUE, "from Part 4 do: ", math.floor(f) % 6 + 32, bcolors.ENDC)

HackathonQuestionLottery()
```

הפלט: שאלות שנבחרו לפי האלגוריתם

```
C:\Users\neriy\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/neriy/PycharmProjects/pythonProject2/main.py
For points - [[2.2, 2.3], [0.0, 0.1]] And x value- 3 We get - 3.0999999999999996
For points - [[2.2, 2.3], [0.0, 0.1]] And x value- 3 We get - 3.0999999999999996
For points - [[2.2, 2.3], [0.0, 0.1]] And x value- 1 We get - 1.0999999999999999
For points - [[5.8, 7.1], [9.4, 0.4]] And x value- 5 We get - 8.588888888888889
For points - [[5.8, 7.1], [9.4, 0.4]] And x value- 4 We get - 10.45
For points - [[5.8, 5.6], [5.1, 4.1]] And x value- 20 We get - 36.02857142857141
Returned Values -> 3.0999999999999996 3.0999999999999996 1.0999999999999999 8.588888888888889 10.45 36.02857142857141
from Part 1 do: 4
from Part 2 do: 14 And 12
from Part 3 do: 28 And 30
from Part 4 do: 32

Process finished with exit code 0
```