מגיש: אביחי חדד(209286665)

**תרגיל 1:** קוד:

# 1-

def most\_common\_letter(string):

""" Function most\_common\_letter gets a string,

and returns the most common letter in that string

"""

# counters\_list

letter\_counter = [0] \* 26

for char in string:

# checking if in range of the lower\upper case letters in ascii.

if 64 < ord(char) < 91 or 96 < ord(char) < 123:

# the index of the letter in alphabet order.

char\_index = ord(char.lower()) - 97

letter\_counter[char\_index] += 1

# the index of the most common letter in the counters list.

max\_index = letter\_counter.index(max(letter\_counter))

# if no letters exists in "string".

if letter\_counter[max\_index] == 0:

return -1

# converting to letter.

most\_common\_letter = chr(max\_index + 97)

return most\_common\_letter

def main():

while True:

string = input("Enter a string please: ")

if string == "quit" or string == "Quit":

print("Thank you for exploring strings and complexity")

break

print("mode('%s') returns %s" % (string, most\_common\_letter(string)))

main()

חלק ב'

חלק א'

פלט:

|  |
| --- |
|  |
|  |

**תרגיל 2:**

1. מהי סיבוכיות הזמן של הפונקציה? נמקו. אם הנכם נעזרים בפרמטר כלשהו, הסבירו מה הוא.

סיבוכיות של O(n), משום שהפונקציה עוברת על כל הרשימה(\*אם לדייק, n-1) פעם אחת מהסוף, ובכל איטרציה בודקת אם האיבר מימין לנוכחי גדול מהאיבר במיקום האינדקס.

1. מהי סיבוכיות הזמן של הפונקציה? נמקו. אם הנכם נעזרים בפרמטר כלשהו, הסבירו מה הוא.

סיבוכיות של O(n^2) משום שידוע לנו לפי סעיף קודם שהפונקציה " insert\_into\_sorted" היא עם סיבוכיות של o(n) לכן אם "insertion\_sort" מריצה את הפונקציה ב-א' n פעמים אז הסיבוכיות היא O(n\*n) = O(n^2).

* 1. עקבו, בעזרת טבלת מעקב, אחר תהליך המיון אשר מבצעת הפונקציה insertion\_sort בעזרת הפונקציה insert\_into\_sorted עבור הרשימה הבאה: a=[5, 4, 3, 1, 2].

**Insertion\_sort(**a=[5, 4, 3, 1, 2]**,** 5):

**i = 1:**

Insert\_into\_sorted(a=[5, 4, 3, 1, 2], 1) ----> a=[4, 5, 3, 1, 2]

**i = 2:**

Insert\_into\_sorted(a=[4, 5, 3, 1, 2], 2) ----> a=[3, 4, 5, 1, 2]

**i = 3:**

Insert\_into\_sorted(a=[3, 4, 5, 1, 2], 3) ----> a=[1, 3, 4, 5, 2]

**i = 4:**

Insert\_into\_sorted(a=[1, 3, 4, 5, 2], 4) ----> a=[1, 2, 3, 4, 5]

**תרגיל 3:** קוד:

# 3-

def maxSort(numbers\_list):

""" Function maxSort gets an list of numbers,

and returns a sorted list, by moving the biggest

number in every iteration to the end of the list.

"""

# running from the end to the beginning.

for i in range(len(numbers\_list) - 1, 0, -1):

maxIndex = i

for j in range(i):

if numbers\_list[j] > numbers\_list[maxIndex]:

maxIndex = j

# replacing elements in list by each other,

# can be done also by doing a,x = x,a but it's less readable.

tmp\_number = numbers\_list[i]

numbers\_list[i] = numbers\_list[maxIndex]

numbers\_list[maxIndex] = tmp\_number

return numbers\_list

חלק ב'

חלק א'

def main():

while True:

list\_size = int(

input("Please enter the number of elements in the list: "))

if list\_size == 0:

print("Thank you for exploring max sort")

break

num\_list = []

# inserting the inputted numbers to the "num\_list".

for i in range(list\_size):

num\_list.append(int(input()))

print("the list, before sorting:", num\_list)

print("the list, after sorting:", maxSort(num\_list))

main()

1. מהי סיבוכיות הזמן של פונקצית המיון, maxSort? O(n^2) משום שמדובר בלולאה מקוננת.

פלט:

|  |
| --- |
|  |