מגיש: אביחי חדד(209286665)

**תרגיל 1:** קוד:

# checking if s1[i] is equal to the last char in "s2"

# (reversed order).

elif s1[i] == s2[-1]:

# setting the index to the next char from right.

index\_of\_s2 = -2

# running in loop to check every char in s2

# is exists in s1(in reversed order).

for j in range(i+1, len(s1)):

if s1[j] == s2[index\_of\_s2]:

# checking is all chars in s2 are checked.

if index\_of\_s2 == -len(s2):

return True

index\_of\_s2 -= 1

return False

def main():

string = "Computer Science"

list\_of\_hidden = ["optic", "nirto", "option"]

for element in list\_of\_hidden:

if hidden\_str(string, element):

print(element, "is hidden in", string)

else:

print(element, "is not hidden in", string)

main()

חלק ב'

# 1-

def hidden\_str(s1, s2):

""" Function hidden\_str gets two strings,

and returns if s2 is hidden in s1.

"""

if len(s2) > len(s1):

return False

index\_of\_s2 = 0

for i in range(len(s1)):

# checking if s1[i] is equal to the first char in "s2"

# (from left to right).

if s1[i] == s2[index\_of\_s2]:

index\_of\_s2 += 1

# running in loop to check every char in s2

# is exists in s1.

for j in range(i+1, len(s1)):

if s1[j] == s2[index\_of\_s2]:

# checking is all chars in s2 are checked.

if index\_of\_s2 == len(s2) - 1:

return True

index\_of\_s2 += 1

חלק א'

פלט:

|  |
| --- |
|  |

**תרגיל 2:** קוד:

חלק ב'

חלק א'

# 2-

def my\_replace(s1, sub):

""" Function my\_replace gets a string and sub string,

and returns a new string with upper-case in every appearance

of the sub in the main string.

"""

if len(sub) > len(s1):

return s1

replaced\_string = ""

index\_of\_sub = 0

index\_of\_s1 = 0

# running in a loop of the indexes of the s1.

while index\_of\_s1 < len(s1):

# checking if both chars in s1 and sub are equal.

if s1[index\_of\_s1] == sub[index\_of\_sub]:

# checking if there is only one letter in sub.

if len(sub) == 1:

replaced\_string += sub.upper()

index\_of\_s1 += 1

continue

index\_of\_sub += 1

# running in loop len(sub) times (maximum...)

while index\_of\_sub < len(sub):

# checking if both chars in s1 and sub are equal,

# or there is only one letter in sub.

if s1[index\_of\_s1 + index\_of\_sub] == sub[index\_of\_sub]:

# checking if we checked every letter in sub according

# to s1 are equal.

if index\_of\_sub == len(sub) - 1:

replaced\_string += sub.upper()

# incrementing "index\_of\_s1" by "index\_of\_sub" in order

# to skip the unwonted iterations in the first while.

index\_of\_s1 += index\_of\_sub + 1

# resetting the variable.

index\_of\_sub = 0

break

index\_of\_sub += 1

else:

# adding to "replaced\_string" the checked letters

# by slicing "s1"

replaced\_string += \

s1[index\_of\_s1:index\_of\_s1 + index\_of\_sub + 1]

# incrementing "index\_of\_s1" by "index\_of\_sub" in order

# to skip the unwonted iterations in the first while.

index\_of\_s1 += index\_of\_sub + 1

# resetting the variable.

index\_of\_sub = 0

break

else:

# adding the letter to "replaced\_string"

replaced\_string += s1[index\_of\_s1]

index\_of\_s1 += 1

return replaced\_string

def main():

while True:

text = input("Enter text: ")

sub\_string = input("Enter substring: ")

# checking if to stop the unless loop

if sub\_string == "" or text == "":

print("Finish")

break

replaced\_text = my\_replace(text, sub\_string)

print(replaced\_text)

main()

פלט:

|  |  |  |
| --- | --- | --- |
| הרצה | **str** | **substr** |
| **1** | In this Question we exercise strings. | is |
| May you have a Pleasant and successful work. | You |
| מחרוזת קלט ריקה |  |
|  | | |
| **2** | Today is a nice day | day |
| Replace occurrence of 'r' | R |
| The sub string is empty | מחרוזת קלט ריקה |
|  | | |