*אביחי חדאד – 209286665*

*אלעד פישר – 318882800*

*בעיה 1 – my\_split()*

*קוד:*

def my\_split(str, ch):

""" Function my\_split gets a string and char,

and returns a list of substrings, and the char as a

divider(the divider doesn't included).

"""

new\_list = []

new\_string = ""

# running on each char in the "str"

for char in str:

if char != ch:

# if chars are not equal concatenating the char to "new\_string"

new\_string += char

else:

# if equal, adding "new\_string" to the list.

new\_list.append(new\_string)

new\_string = ""

# adding the last "new\_string"

new\_list.append(new\_string)

return new\_list

def main():

""" Function main calls the "my\_place" function

and prints the result according to hard-codded string.

"""

string = "This is an example of how split works"

print(my\_split(string, " "))

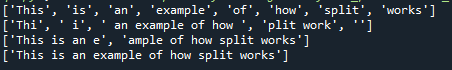
print(my\_split(string, "s"))

print(my\_split(string, "x"))

print(my\_split(string, "y"))

main()

*פלט לבעיה 1:*



*בעיה 2 –is\_legal ip*

**תתי-משימות:**

1. פיצול המחרוזת בהתאם לנקודות שמופיעות בה.
2. לבדוק אם כל מספר לאחר פיצול מכיל אפסים מובילים או לא.
3. לבדוק שקיים רק ארבעה נקודות(כלומר המחרוזת מתפצלת ל-4).
4. לבדוק האם המחרוזת מכילה רק מספרים בין 0 ל-255.

קוד:

def has\_leading\_zeros(number):

""" Function has\_leading\_zeros gets a number

and returns if the number has\_leading\_zeros

(expect zero as one digit).

"""

return len(number) > 1 and number.startswith("0")

def is\_legal\_ip(string):

""" Function is\_legal\_ip get a string

and returns if the string is legal ip or not.

"""

# splitting the string by "." into a list.

split\_string = string.split(".")

# checking if we have 4 decimal number.

if len(split\_string) != 4:

return False

# running on each element in the "split\_string".

for element in split\_string:

# checking if the element is valid.

if element.isdigit() and 0 <= int(element) <= 255 \

and not has\_leading\_zeros(element):

continue

else:

# if not valid.

return False

return True

def main():

""" Function main calls "is\_legal\_ip" function

with hard-codded strings and prints the returned result.

"""

print(is\_legal\_ip("192.168.1.1"))

print(is\_legal\_ip("125.34.251.43"))

print(is\_legal\_ip("001.23.45.123"))

print(is\_legal\_ip("125.512.100.xy8"))

print(is\_legal\_ip("125.512.."))

print(is\_legal\_ip("192.168.0.1"))

main()

פלט לבעיה 2:



בעיה 3 – "שרשרת מקטעים":

קוד:

def strCompress(string):

""" Function strCompress gets a string and returns its compressed form.

"""

compressed\_str = ""

# declaring a counter the will holds the number of

# appearances of the char.

counter = 0

for i in range(len(string) - 1):

counter += 1

# checking if the current char isn't equal the next one.

if string[i] != string[i+1]:

# concatenating the char and "counter" and adding it

# to "compressed\_str"

compressed\_str += string[i] + str(counter)

# resetting counter

counter = 0

# adding the last char to the "compressed\_str".

compressed\_str += string[-1] + str(counter + 1)

return compressed\_str

def strRestored(string):

""" Function strRestored gets a compressed string

and returns the restored form of the string.

"""

restored\_str = ""

multiplier = ""

index = 0

# running loop len(string) times.

while index < len(string):

if string[index].isalpha():

# declaring "alpha" that holds the char.

alpha = string[index]

# running loop until the string[i] is letter.

for i in range(index + 1, len(string)):

if string[i].isalpha():

# changing index to skip the unwonted

# iterations in the while loop.

index = i - 1

break

multiplier += string[i]

# adding the restored char to the "restored\_str"

# by multiplying in counter.

restored\_str += alpha \* int(multiplier)

multiplier = ""

index += 1

return restored\_str

def main():

""" Function main gets an input from the user of

section chain and compressed string and prints the

returned value from the right function.

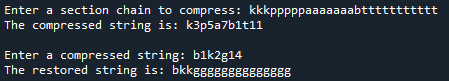
"""

section\_chain = input("Enter a section chain to compress: ")

print("The compressed string is:", strCompress(section\_chain))

compressed\_string = input("Enter a compressed string: ")

print("The restored string is:", strRestored(compressed\_string))

main()

פלט: