Q1

Strings Methods: split, find, replace

```
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64
bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> "xyzw".split("yz")
['x', 'w']

>>> "xyzw".find("w")
3

>>> "xyzw".replace("x","a")
'ayzw'

>>> str.split("xyzw","yz")
['x', 'w']

>>> str.find("xyzw","w")
3

>>> str.replace("xyzw","x","a")
'ayzw'
```

Lists Methods: pop, append, insert

```
>>> ["x","y","z","w"].pop(0)
'x'
>>> list.pop(["x","y","z","w"],0)
'x'


>>> t=["x","y","z","w"]
>>> t.append("A")
>>> t
['x', 'y', 'z', 'w', 'A']

>>> t= ["x","y","z","w"]
>>> list.append(t,"A")
>>> t
['x', 'y', 'z', 'w', 'A']

>>> r=["x","y","z","w"]
>>> r.insert(1, "A")
>>> r
['x', 'A', 'y', 'z', 'w']

>>> r=["x","y","z","w"]
>>> list.insert(r,1,"A")
>>> r
['x', 'A', 'y', 'z', 'w']
```

Q2

עבור הקלט "87654321"

Iteration	i	ID[i]	val	Total
1	0	8	8	8
2	1	7	7	13
3	2	6	6	19
4	3	5	5	20
5	4	4	4	24
6	5	3	3	30
7	6	2	2	34
8	7	1	1	36

Q3

סעיף א':

	2**100	2**300	2**700	2**1500
Sol 1	8.79*10^(-6) sec	1.94*10^(-5) sec	4.84*10^(-5) sec	0.0014 sec
Sol 2	3.49*10^(-6) sec	6.2*10^(-6) sec	1.27*10^(-5) sec	2.64*10^(-5) sec

הסבר קצר לתוצאות: אם נתייחס לגידול **באורך** הקלט, נשים לב שמקלט אחד למשנהו הוא גדל בערך פי 2. בהתאם לכך, אכן זמני הריצה גדלים ביחס דומה, כלומר – פי 3 לפי הפתרון הראשון, ופי 2 לפי הפתרון השני. (הגידול הוא לינארי)

סעיף ב':

	2**100	2**300	2**700	2**1500
Sol 3	5.4*10^-6 sec	6*10^-6 sec	8.3*10^-6 sec	1.46*10^-5 sec

ניתן להבחין מתוך זמני הריצה כי אכן הפתרון השלישי יעיל יותר מהשניים האחרים.

סעיף ג:

להלן טבלה המפרטת את זמני הריצה עבור שני קלטים באורך 1000. הראשון הוא אלף פעמים הספרה "1", והשני הוא הספרה "1" עם 999 פעמים הספרה "0":

	0 zeros	999 zeros
sol1	17e-04	17e-04
sol2	1e-04	1.4e-04
sol3	5.63e-05	5.65e-05

ניתן לראות כי זמני הריצה כמעט זהים. כלומר לכמות האפסים בקלט בפועל אין השפעה משמעותית (עד כדי לא קיימת) על זמן הריצה. ניתן לשער כי ההסבר לזה הוא שאופן הריצה אינו מושפע מהמצאות או אי-המצאות של "0" בקלט, וכמות האיטרציות, ולכן גם זמן הריצה, יהיו דומים עבור קלטים באותו אורך.

סעיף ד':

הרצת בדיקה של 2**25 לקחה 3 שניות. 2**26 ארכה 6 שניות, 2**27 ארכה 12 שניות.

באופן אינדוקטיבי נשער כי הזמן שיקח ללולאה להתרחש עם קלט של 2**1000 יהיה (2**975)*(3) שניות. זה שקול ל: **9.580033e+293** שניות.

ניתן לשער מאחר שכמות האיטרציות היא הערך של **num**, זמני הריצה יגדלו באופן לינארי ביחס ל-**num**. ההבדל הוא שבסעיף ד' כמות האיטרציות הינה ביחס ישר לערך הקלט, בעוד שבסעיף א' מדובר על אורך הייצוג העשרוני (כמספר הספרות) של הקלט שקטן בהרבה מהערך עצמו.

Q4-a

היינו מגדילים את המערך: היינו מוסיפים למשתנה char את המספרים 0-9 ולולאת ה-for הייתה בודקת גם את המספרים, אם הם נמצאים בתוך text . בכל פעם שכן זה היה מתסווף למונה שלנו cnt .