



1.

הקוד:

- יוצר רשימה ריקה ששמה res

- מפצל את כל המחרוזות my_str לפי הרווחים, מכניס כל מחרוזת (למשל "The", "first" וכו'...) לרשימה לפי הסדר ושומר את הרשימה ב-my_str.

- שומר במשתנה first את הערך 1, במשתנה second את אורך הרשימה my_str (שהוא 6, כמספר המילים במחרוזת המקורית) ושומר במשתנה third את הערך 2.

- נכנס ללולאה בה עוברים על המילים במקום הזוגי במחרוזת ומוסיפים אותה לסוף הרשימה שבסוף נחזיר.

לסיכום, הקוד לוקח את המחרוזת my_str ומחזיר רשימה של המילים במקומות הזוגיים בה, כלומר רשימה עם המילה השניה, המילה הרביעית וכו'...

Oneliner שקול לכל הקוד הזה הוא:

```
res = my_str.split()[1::2]
```

2.

הקוד:

- מאתחל מילון ריק
 - נכנס ללולאה עבור כל המספרים בסדרה 0, 94, 97, 100, ... (נסמן ב-x).
 - עבור כל x שמתחלק ב-n נוסף איבר למילון שהמפתח שלו הוא x והערך שלו הוא "{x} is divided by {n}.\n"
 - ועבור כל x **שלא** מתחלקים ב-n נוסף איבר למילון שהמפתח שלו הוא x והערך שלו הוא "the remainder of {x} is: {x%n}.\n"
 - מדפיס את המילון.
- לסיכום, קוד זה מדפיס עבור כל המספרים 0, 94, 97, 100, ... האם הם מתחלקים ב-n מסוים ואם לא אז שארית החילוק מודפסת.

Oneliner שקול לכל הקוד הזה הוא:

```
print(*(f"{x} is divided by {n}.\n" if x % n == 0 else f"the remainder of {x} divided by {n} is: {x % n}.\n" for x in range(100, 0, -3)))
```

3.

הקוד עובר בלולאה על טווח מ-0 עד ל-ord('z') (שערכו 122 והוא המקסימלי מבין השלושה), כלומר 123 איטרציות, בהן אנחנו בודקים מה התו המתאים למספר שאנחנו כרגע עליו בטווח (נגיד i), ואם הוא אות מהאלפבית הלועזי או מספר אנחנו נדפיס שורה שמציגה את i וגם את האות שהוא מייצג ב-ASCII.

Oneliner שקול לכל הקוד הזה הוא:

```
print(*(f"The ASCII number {i} represent the char '{chr(i)}'" for i in range(0, 123) if chr(i).isalpha() or chr(i).isdigit()), sep="\n")
```

4. הקוד עובר בלולאה על רשימה של מספרים וממיר כל אחד מהם לתו שהוא מייצג ב-Unicode. לאחר ההמרה, הוא משרשר את התו למחרוזת שמתחילה ריקה, לכן בסך הכל הקוד מדפיס את ה-Unicode של רשימת מספרים כך שכל איבר ברשימה מייצג אות.

```
print("".join(chr(num) for num in list_c))
```