



3.

1.

קטע קוד זה מפצל את `my_str` למערך של מילים, ולוקח כל מילה במקום האי זוגי ומכניס אותה למערך `res`. הפלט יהיה :

```
['first', 'is', 'easiest!']
```

Oneliner שקול :

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

תחילה אנו מפצלים את השורה למילים, ולאחר מכן מבצעים slicing אשר מתחיל מהמילה במקום 1, עד סוף המערך וקופצת כל פעם ב-2.

2.

קטע קוד זה מגדיר מילון שהמפתחות שלו הם המספרים בין 100 ל-0 בקפיצות של מינוס 3, כלומר 100,97,94 וכו'.

עבור כל מפתח, הערך יהיה המחזורת האם X מתחלק ב-N, אם לא אז מה השארית. לבסוף, מודפסים כל הערכים של המילון, לפי הסדר. מכיוון שהתחלנו מ-100 וירדנו ל-0, אז כך גם ההדפסה תהיה בהתאם. השימוש ב-* בהדפסה נועדה שיודפסו הערכים בהפרדה עם רווחים, ולא יודפסו הערכים של המילון כרשימה אחת.

נשים לב שבמילון לא יופיעו מספרים שהם לא בקפיצות של מינוס 3 בין 100 ל-0.

Oneliner שקול :

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

אנו רצים על כל מספר בין 100 ל-0 בקפיצות של מינוס 3, ובהתאמה מדפיסים את הנדרש. נשים לב שהורדנו את \n בסוף כל הדפסה מכיוון שפעולה זו קוראת אוטומטית בשימוש print, להבדיל מהדפסת ערכי המילון שלא מוסיפה ירידת שורה.

3.

קטע קוד זה מדפיס את המספר הASCII ואת התו הASCII עבור כל תו שהוא מספר או אות בטבלה הASCII. מכיוון ש'z' היא האות עם המספר הASCII הגבוה ביותר, הלולאה תרוץ מ-0 עד המספר שלה -122 ועוד 1, כלומר תרוץ עד 123.

עבור האיטרציות 0-47 לא יודפס דבר כי בטבלה אין מספרים או אותיות. לאחר מכן יודפסו כל המספרים מ-0-9, כל האותיות הגדולות בסדר אלפבתי וכל האותיות הקטנות בסדר אלפבתי.

Oneliner שקול :

```
[print(f"The ASCII number {i} represent the char '{chr(i)}'") for i in range(0, max(ord('9'), ord('z'), ord('Z'))+1) if chr(i).isalpha() or chr(i).isdigit()]
```

4.

קטע קוד זה ממיר רשימה של מספרים לאותיות לפי הערך הASCII שלהם, ומשרשר אותם למחרוזת. לבסוף המחזורת מודפסת.

הפלט יהיה :

```
Python is fun!
```

Oneliner שקול :

```
print(''.join([chr(i) for i in list_c]))
```

אנו יוצרים מערך של אותיות מהמספרים ולאחר מכן מפעילים עליו את פקודת join עם הפרדה ריקה. כל זה מודפס כנדרש.