

1

הסביר על הקוד: הקוד לוקח את המחרוזת ומפצל אותה לרשימה של מילים נפרדות. אחר כך, בעזרת לולאה, אנחנו אוספים לרשימה חדשה רק את המילים שנמצאות במיקומים אי-זוגיים - ככלומר מתחילה מהמילה השנייה (אינדקס 1) וקופצים ב-2 בכל פעם.

קוד חלופי בשורה אחת:

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

2

הסביר על הקוד: רצים בלולאה יורדת מ-100 ועד 1 בקפיצות של 3. עבור כל מספר, בודקים אם הוא מחלק ב-3 בלי שארית ושומרים במלון משפט שמתאר את התוצאה (מתחלק או מה השארית). בסוף, בעזרת תוכנית לפרק איברים, מדפיסים את כל המשפטים שנאספו במלון ברצף ולא רוחים ביניהם.

קוד חלופי בשורה אחת:

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

3

הסביר על הקוד: הלולאה עוברת על כל המספרים מ-0 ועד לערך ה-ASCII הכי גבוהה מבין התווים שניתנו. בכל שלב נבדק אם המספר מסוים הזה יכול להתרגם לאות חוקית באנגלית או לספירה. אם התנאי מתקיים, מדפיסים את המספר עצמו יחד עם התו שהוא מייצג.

קוד חלופי בשורה אחת:

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

4

הסביר על הקוד: עוברים על רשימה נתונה של מספרים שלמים. כל מספר הופך בתורו לטו ASCII המתאים לו, ואנחנו לחברים אותו אחד לשני כדי לבנות מחרוזת אחת שלמה שאוגרת הכל. כשהלולאה מסתיימת, אנחנו פשוט מדפיסים את הטקסט הסופי שהתקבל.

קוד חלופי בשורה אחת:

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