

1

הסבר 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 - כלומר מתחילים מהמילה השנייה (אינדקס 1) וקופצים ב-2 בכל פעם. קוד חלופי בשורה אחת:

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

2

הסבר על הקוד: רצים בלולאה יורדת מ-100 ועד 1 בקפיצות של 3. עבור כל מספר, בודקים אם הוא מתחלק ב-n בלי שארית ושומרים במילון משפט שמתאר את התוצאה (מתחלק או מה שארית). בסוף, בעזרת כוכבית לפריקת איברים, מדפיסים את כל המשפטים שנאספו במילון ברצף וללא רווחים ביניהם. קוד חלופי בשורה אחת:

```
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)], sep="")
```

3

הסבר על הקוד: הלולאה עוברת על כל המספרים מ-0 ועד לערך ה-ASCII הכי גבוה מבין התווים שניתנו. בכל שלב נבדק אם המספר המסוים הזה יכול להיתרגם לאות חוקית באנגלית או לספרה. 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

הסבר על הקוד: עוברים על רשימה נתונה של מספרים שלמים. כל מספר הופך בתורו לתו ה-ASCII המתאים לו, ואנחנו מחברים אותם 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]))
```