

```
new_str = [ord(elem) for elem in my_str] (1)
```

שומרת את הערך המספרי unicode code של כל תו במחרוזת my_list לפי טבלת (unicode table) באמצעות הפונקציה ord שמקבלת תו ומחזירה את הערך המספרי המתאים.

```
print(*{x:f"{x}^{x} is {x ** x}\n" for x in range(10,0,-1)}.values()) (2)
```

עובר מהסוף להתחלה על רשימה שמכילה מספרים שמסודרים מהקטן לגדול (1-10), בקפיצות של 1- (כלומר מהסוף עד להתחלה בקפיצות של 1), ויוצר מילון שבו כל ערך מהרשימה הוא המפתח, והערך של מפתח שלו זו מחרוזת שמעידה על הערך המתקבל מהעלאת המספר בריבוע, בין כל זוג מפתח וערך יש ירידת שורה, לבסוף מדפיסים את הערכים של המפתחות עם אופרטור פיזור שיוצר סדרת ערכים (בלי הסוגריים) כך שמודפסים לאחר מכאן, כל אחד בשורה.

```
print(*[f"The ASCII number {i} represent the char {chr(i)}\n" for i in range(ord('A'), ord('z')+1, 2) if chr(i).isalpha()]) (3)
```

עוברים בלולאה for מהערך המספרי של התו 'A' לפי unicode table על ידי פונקציה ord ועד לערך הגדול מאחד מהערך של התו z בקפיצות של 2 על ידי שימוש ב range כך שכל פעם אם הערך המספרי מתאים לתו מהאלפאבית אנחנו מדפיסים הודעה שמעידה על המספר שמייצג את התו.

```
print("".join([chr(num) for num in list_c])) (4)
```

מאתחילים מחרוזת ריקה ועוברים על הרשימה הנתונה של המספרים בלולאה for, מוסיפים למחרוזת את התו שמיוצג על ידי המספר לפי (unicode table), לבסוף מדפיסים את המחרוזת.