

# מבוא לתכנות מערכות (02340124) –

## תרגיל בית 5

מגישים:

- ברק מטרי, ת.ז: 212269278
- נועה כספי, ת.ז: 212153910

Q.5.3. (רצו) תיחזקו



### Q.3.1

הקוד מפצל את *my\_str* לרשימת מילים  
(*my\_str*.split()) , ואז מוסיף ל*res* את המילים  
באינדסים ... 1,3,5 (מתחיל ב-1 *first* , קופץ ל-2 *third* =  
עד שמגיע ל-*len(my\_str)* .  
על כן, *res* נהיה כל מילה שניה, החל מהמילה השניה במחרוזת.  
עבור הדוגמא הנתונה:

```
my_str = "The first one is the easiest!"
```

הרשימה המפוצלת המתקבלת היא:

```
["The", "first", "one", "is", "the", "easiest!"]
```

ו-*res* הופך ל:

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



כמו כן,

***oneliner* שקול הוא:**

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

### Q.3.2

קטע הקוד הנתון מבצע איטרציה לפי *x* בטווח  
(-3, 0, 100)  $\Leftarrow$  *range*(100, 0, -3) כל פעם מ-100 עד לערך  
החיובי האחרון אליו מגיע (... ,94,97,100).  
לאחר מכן, עבור כל *x* , מאחסנים *string* ב-*my\_dictionary[x]*  
באופן הבא:

- אם  $x \% n == 0$ , כלומר שארית החלוקה במשתנה  $n$  היא 0  
אז מודפס `"{x} is divided by {n}.\n"`.

- אחרת, יודפס:

`"the remainder of {x} divided by {n} is: {x \% n}.\n"`

ולבסוף, התכנית מדפיסה את כל ערכי המילון ששורשרו (משום ש-  
`print(* values, sep = "")` מדפיס כל ערך בלי  
הפרדה/seperator).

בנוסף, `dict.values()` משמר סדר הכנסה, אז הסדר שיוודפס  
יתאים לסדר בלולאה.

הקוד שבדוגמא, ידפיס:



לכן,

***oneliner* שקול 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)], sep = "")
```

Q.3.3

`ord(...)` למשל `ord(c)` - ממיר את התו  $c$  ל-`integer` (לפי  
`chr(i)` ו-`ASCII/Unicode` ממיר `integer` "i" לתו עם אותו קוד.

לכן:

`max(ord('9'), ord('z'), ord('Z'))`

שקול ל:

$$\text{ord}('z') = 122$$

אז הלולאה מריצה  $i = 0, 1, 2, \dots, 122$ . כאשר:

לכל  $i$ , הקוד בודק אם  $\text{chr}(i)$  הוא האות  $(\text{isalpha}())$  או ספרה  $(\text{isdigit}())$ .

אם מתקיים (כלומר מחזיר  $\text{True}$ ), אז הקוד ידפיס את השורה:

The ASCII number  $i$  represent char ' $X$ '

ועל כן, ההדפסה ממפת את:

- הספרות 0 – 9 (קוד ASCII: 48 – 57).
- אותיות גדולות/uppercase, A – Z (קוד ASCII: 65 – 90).
- אותיות קטנות/lowercase, a – z (קוד ASCII: 97 – 122).

בסדר עולה (לפי  $i$ ).



כמו כן, ניתן להשתמש ב- $\text{isalnum}()$  (אשר שקול

ל- $(\text{isalpha}() \text{ or } \text{isdigit}())$  בטווח זה)



לפיכך,

**נקבל את ה-oneliner השקול:**

```
print('\n'.join(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).isalnum())
```

Q.3.4

הקוד הנתון מאתחל את המחרוזת הריקה `tmp_chr = ""` מבצע איטרציה מעל כל `integer num` ב-`list_c`, ממיר כל `integer` אל האות (`character`) שלו, בעזרת `chr(num)`, ולאחר מכן, משרשר זאת ל-`tmp_chr`.

לבסוף, מדפיס את המחרוזת הסופית המתקבלת.



כלומר, הקוד מפענח רשימה של קודים ב-`ASCII` אל מחרוזת אחד, ומדפיס אותה.

עבור הרשימה המתוארת בדוגמא:

[80,121,116,104,111,110,32,105,115,32,102,117,110,33]

הקוד ידפיס:

[*Python is fun!*]



לכן (בעזרת `join`),

<p><b><i>oneliner</i> שקול הוא:</b></p> <pre><i>print(''.join(map(chr, list_c)))</i></pre>
--