

## Dry Part

לצורך קטע הקוד הבא, מוגדר עבורכם משתנה בשם `my_str` המכילה מחוץ למשל: .1

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

הסבירו מה הקוד הבא עושה וכותבו oneliner שקול לכך שהמשתנה `res` יוכל את אותה התוצאה:

```
res = []
my_str_ = my_str.split()
first, second, third = 1, len(my_str_), 2
while first < second:
    res.append(my_str_[first])
    first += third
```

The code turns `my_str` into a list that represents the original string, where each word is a separate item, and adds each word that has an odd index to `res`.

In our example: `res = [first, is, easiest!]`

**Oneliner:**

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

. לצורך קטע הקוד הבא, מוגדר עבורכם המשתנה  $n$  (למשל  $n=42$ ) , הסבירו מה הקוד הבא עושה וכותבו one liner שקול לכך שקטע הקוד השקול יופיע למסך את אותה התוצאה בדיקות:

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

שימוש לב – אין משמעות לסדר בו מודפסים האיברים , השתמשו בשיטת הדפסה בקטע הקוד המצורף וכיום לב שניתן לפצל את השורה עם היא אורך לצורך קריאות) .

The code creates a dict in the following way:

Keys: The numbers 100,97,94,...,1

Values: for each key, a string that specifies if it's divisible by  $n$ , and if it's not, the string will specify its remainder.

In the end, all the strings are printed.

#### Oneliner:

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

.3 קראו על הפקנציות **ord**, **chr** ב `python`. הסבירו מה הקוד הבא עושה וכתבו oneliner שקול בר שחר הדפסה:  
זהה זהה:

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

According to ASCII tables:

`ord('9')` = 57

`ord('z')` = 122

`ord('Z')` = 90

so the range in the first line of the code is `range(0,123)`

The code goes through all the characters with the ascii values of 0 to 123, which includes all the letters and all the digits.

For every letter or digit, the code prints a message.

The digits will be printed first, then the uppercase letters, then the lowercase letters.

### Oneliner:

```
[print(f"The ASCII number {i} represent the char '{chr(i)}'") for i in range(0,123) if
chr(i).isalpha() or chr(i).isdigit()]
```

.4. לצורך קטע הקוד הבא, מוגדר עבורם משתנה בשם `list_c` המכיל רשימה של מספרים שלמים, למשל:

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

הסבירו מה הקוד הבא עושה ובתו oneliner שקו אחד שההדפסה תהיה זהה:  
שיםו לב שהקוד צריך להציג את אותה התוצאה לכל מערך רנדומלי של מספרים, המערך הנוכחי הוא רק דוגמה.

```
tmp_chr = ""  
for num in list_c:  
    tmp_chr += chr(num)  
print(tmp_chr)
```

רמז – קראו על המתודה `join` לערכים מסוג `string`, תוכלו להשתמש בה לצורך הפתרון.

The code goes through `list_c` and converts each number into the ASCII char it represents, and creates a string from these chars. It then prints the string. If `list_c` contains an element that is not a valid ASCII value, there will be an error.

#### Oneliner:

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