

תרגול מקיף – מלולאות ועד עבודה עם קבצים

חוברת תרגול נלווית למצגות של יחידות 7-15. בכל יחידה השאלות מסודרות מהקל למאתגר: קודם **מה יודפס? ואיתור באגים** (פותרים על הנייר, עם טבלת מעקב), ואחר כך **כתבו תוכנית**. בסוף החוברת יש מפתח תשובות עם פתרון לדוגמה לכל משימה. כל הקוד בחוברת הורץ, והפלט במפתח הוא הפלט האמיתי.

Tuple • 12

פונקציות • 11

מיון • 10

רשימות • 9

8 • אופרטורים לוגיים וסיביים

7 • לולאות

15 • קבצים מתקדם

14 • קבצים

13 • מילונים

113 שאלות • 9 יחידות • קל → בינוני → מאתגר

7 לולאות: for, while ופירוק מספר לספרות שאלות 15

מתחילים במעקב על הנייר: לכל לולאה בונים טבלת מעקב עם עמודה לכל משתנה ועמודה לתנאי. אחר כך כותבים לולאות בעצמכם.

תזכורת

- `for i in range(start, stop, step)`: ה-`stop` לא נכלל. `range(5)` נותן 0 עד 4.
- `while` תנאי: רצה כל עוד התנאי `True`. חייבים לשנות משהו בתוך הלולאה, אחרת היא אינסופית.
- צובר: `total = 0` לפני הלולאה ו-`total += x` בתוכה. מונה: `count += 1`.
- זקיף: קולטים לפני הלולאה, ובסוף כל סיבוב קולטים שוב. ערך הזקיף עצמו לא נספר.
- פירוק לספרות: `n % 10` היא ספרת האחדות, ו-`n //= 10` מוחק אותה.
- `break` יוצא מהלולאה, `continue` מדלג לסיבוב הבא.

קל

1 מה יודפס?

```
for i in range(1, 6):
    print(i * 2, end=" ")
```

קל

2 מה יודפס? שימו לב לצעד השלילי.

```
for i in range(10, 0, -3):  
    print(i)
```

קל

3 בנו טבלת מעקב עם העמודות `i` , `total` ו- `i < 10` , ואז כתבו מה יודפס.

```
total = 0  
i = 1  
while i < 10:  
    total += i  
    i += 3  
print(total, i)
```

קל

4 מה יודפס? כמה פעמים רצה הלולאה?

```
count = 0  
for ch in "banana":  
    if ch == "a":  
        count += 1  
print(count)
```

בינוני

5 עקבו אחרי `n` , `d` ו- `s` בכל סיבוב.

```
n = 4071  
s = 0  
while n > 0:  
    d = n % 10  
    s += d  
    n //= 10  
print(s)
```

בינוני

6 מה יודפס? (`continue` ו- `break` באותה לולאה)

```
for i in range(1, 10):
    if i % 3 == 0:
        continue
    if i == 8:
        break
    print(i, end=" ")
```

בינוני

7 לולאות מקוננות: מה יודפס?

```
for i in range(1, 4):
    line = ""
    for j in range(i):
        line += str(i)
    print(line)
```

בינוני

8 מה יודפס כשהמשתמש מקליד את המספרים 5, 8, 2 ואז 0?

```
total = 0
count = 0
x = int(input("מספר: "))
while x != 0:
    total += x
    count += 1
    x = int(input("מספר: "))
print(total, count)
```

קלט: 5 , 8 , 2 , 0

בינוני

9 הלולאה אמורה להדפיס 1 עד 5, אבל היא לא נגמרת אף פעם. מה הבאג? (כאן הוספנו בדיקה שעוצרת אחרי 20 סיבובים).

```
i = 1
steps = 0
while i <= 5:
    print(i)
    steps += 1
    if steps == 20:
        raise RuntimeError("infinite loop")
    i += 1
```

בינוני

10 כמה פעמים תרוץ כל לולאה? (א) `for i in range(3, 12, 4)` (ב) `for i in range(5, 5)` (ג) `while x > 0` כש- `x = -2` לפני הלולאה.

קל

11 קלטו מספר שלם חיובי. הדפיסו את מספר הספרות שלו, את סכום הספרות ואת מספר הספרות הזוגיות. (1234 ← 4 ספרות, סכום 10, זוגיות 2.)

בינוני

12 קלטו מספר והדפיסו אותו הפוך כמספר (3720 ← 273), בלי להפוך מחרוזת. רמז: `rev = rev * 10 + d`.

בינוני

13 קלטו ציונים עד שמוקלד -1. הדפיסו את הממוצע, את הציון הגבוה ואת מספר הנכשלים (מתחת ל-55). אם לא הוקלד אף ציון, הדפיסו אין ציונים.

בינוני

14 הדפיסו לוח כפל של 1 עד 5 בעזרת לולאות מקוננות. כל שורה תכיל 5 מספרים מופרדים בטאב (`\t`).

מתגר

15 מספר מושלם שווה לסכום המחלקים שלו (בלי המספר עצמו). לדוגמה $6 = 1 + 2 + 3$. הדפיסו את כל המספרים המושלמים בין 1 ל-1000.

שאלות 12

אופרטורים לוגיים וסיביים

8

and, or ו-not מחברים תנאים. האופרטורים הסיביים &, |, ^, ~, << ו->> עובדים על הביטים של מספר שלם.

- True: and רק אם שני הצדדים True. or: True אם לפחות צד אחד True. not הופך.
- סדר עדיפות: not לפני and, ו-and לפני or. בספק, שמים סוגריים.
- הערכה מקוצרת: ב-False and... ובשורה or True... הצד הימני לא נבדק בכלל.
- `'0b110' ← bin(6)` (וגם), `|` (או), `^` (או מוציא: 1 רק כשהביטים שונים).
- `1 << x` מכפיל ב-2, `1 >> x` מחלק ב-2 (חלוקה שלמה).

קל

1 מה יודפס?

```
x = 7
print(x > 5 and x < 10)
print(x < 5 or x == 7)
print(not x > 5)
```

בינוני

2 מה יודפס? שימו לב לסדר העדיפות.

```
a = True
b = False
c = True
print(a or b and not c)
print((a or b) and not c)
```

בינוני

3 מה יודפס?

```
x = 6
print(x > 5 and x % 2 == 1, not x < 3 or x == 0)
```

בינוני

4 מה יודפס? כתבו קודם את 12 ואת 10 בבינארי.

```
print(bin(12), bin(10))
print(12 & 10, 12 | 10, 12 ^ 10)
```

5 מה יודפס?

בינוני

```
print(5 << 1, 5 << 3, 40 >> 2, 7 >> 1)
```

6 הערכה מקוצרת: מה יודפס? למה אין שגיאת חלוקה באפס?

מתגר

```
x = 0
print(x != 0 and 10 / x > 1)
print(x == 0 or 10 / x > 1)
```

7 כתבו תנאי אחד לכל אחד מהמקרים: (א) `age` בין 13 ל-19 כולל. (ב) `ch` הוא תנועה באנגלית (`u` , `o` , `i` , `e` , `a`). (ג) `year` מעוברת: מתחלקת ב-4 ולא ב-100, או מתחלקת ב-400.

בינוני

8 התנאי אמור לבדוק אם `grade` הוא 90 או 100, אבל הוא מדפיס "מצוין" גם לציון 40. למה? (הקוד לא קורס. כדי לראות את הבאג הוספנו בדיקה.)

מתגר

```
grade = 40
if grade == 90 or 100:
    raise AssertionError("the condition is always True")
```

9 נתון `flags = 0b0101`. מה הערך (בבינארי) אחרי כל פעולה, כשכל פעולה מתחילה מ-`0b0101`? (א) `flags | 0b0010` (ב) `flags & 0b0100` (ג) `flags ^ 0b0001`

בינוני

10 קלטו שלושה צדדים של משולש. הדפיסו `שווה צלעות`, `שווה שוקיים` או `שונה צלעות`. לפני כן בדקו שהמשולש קיים: כל שני צדדים יחד ארוכים מהשלישי. אם לא, הדפיסו `אין משולש`.

בינוני

11 קלטו מספר שלם והדפיסו אם הוא זוגי `בלי %`. רמז: הביט הימני של מספר זוגי הוא 0, ו-`n & 1` מחזיר בדיוק אותו.

בינוני

12 קלטו מספר שלם חיובי והדפיסו כמה ביטים דולקים (1) יש בייצוג הבינארי שלו, בעזרת `& 1` ו-`>>` בלבד. (`13 = 1101` ← 3)

מתגר

אינדקסים, חיתוך, פעולות על רשימה, מעבר בלולאה, איברים סמוכים ורשימת מונים. שני הנושאים האחרונים מודגשים בהנחיות לבחינה של תשפ"ז.

תזכורת

- אינדקס ראשון 0, אחרון -1. `lst[1:3]` מחזיר את האיברים במקומות 1 ו-2 (3 לא נכלל).
- `append(x)` מוסיף לסוף, `insert(i, x)` מכניס במקום `i`, `pop()` מוחק ומחזיר את האחרון, `remove(x)` מוחק את המופע הראשון של `x`.
- `len`, `sum`, `max`, `min`, `x in lst`, `lst.count(x)`, `lst.index(x)`.
- איברים סמוכים: `for i in range(len(lst) - 1):` ומשוים `lst[i]` ל-`lst[i + 1]`.
- רשימת מונים: `counts = [0] * 10`, ובלולאה `counts[value] += 1`. הערך הופך לאינדקס.

קל

1 מה יודפס?

```
nums = [3, 8, 1, 9, 4]
print(nums[0], nums[-1], nums[1:3], nums[:2], len(nums))
```

קל

2 מה יודפס?

```
nums = [3, 8, 1, 9, 4]
nums.append(nums[1] + nums[-1])
nums.pop(0)
nums.insert(1, 7)
print(nums)
```

קל

3 מה יודפס?

```
words = ["cat", "dog", "cat", "bird"]
print(words.count("cat"), words.index("dog"), "fish" in words)
words.remove("cat")
print(words)
```

בינוני

רשימה היא אובייקט אחד עם שני שמות: מה יודפס? 4

```
a = [1, 2, 3]
b = a
c = a[:]
b.append(4)
print(a)
print(c)
```

בינוני

איברים סמוכים: מה יודפס? 5

```
lst = [4, 4, 7, 2, 2, 2, 9]
count = 0
for i in range(len(lst) - 1):
    if lst[i] == lst[i + 1]:
        count += 1
print(count)
```

בינוני

רשימת מונים: מה יודפס? 6

```
digits = [3, 1, 3, 0, 9, 3, 1]
counts = [0] * 10
for d in digits:
    counts[d] += 1
print(counts)
print(counts.index(max(counts)))
```

בינוני

מה יודפס? ותארו במשפט אחד מה הלולאה עושה. 7

```
lst = [5, 2, 8, 1, 9]
for i in range(len(lst)):
    if lst[i] % 2 == 0:
        lst[i] = 0
print(lst)
```

8 מה הבאג?

קל

```
lst = [10, 20, 30]
for i in range(len(lst) + 1):
    print(lst[i])
```

9 מה ההבדל בין `for x in lst:` לבין `for i in range(len(lst)):`? מתי חייבים את הצורה השנייה?

בינוני

קל

10 קלטו מספרים עד שמוקלד 0 ושמרו אותם ברשימה. הדפיסו את הממוצע ואת כל המספרים שגדולים ממנו.

קל

11 כתבו קוד שמוצא את הגדול ביותר ברשימה ואת המיקום שלו, בלי `max`.

מאתגר

12 הרצף הארוך: מצאו ברשימה את אורך הרצף הארוך ביותר של מספרים עולים (כל איבר גדול מהקודם). ב- `[1, 3, 5, 2, 4, 6, 8, 1]` התשובה 4 (2,4,6,8).

מאתגר

13 הטילו קובייה 600 פעמים (`random.randint(1, 6)`) וספרו ברשימת מונים כמה פעמים יצא כל מספר. הדפיסו את התוצאות. בהרצה לדוגמה קבענו `random.seed(1)` כדי שהתוצאה תהיה קבועה.

10 מיון רשימות שאלות 11

`sorted()` ומה המחשב עושה "מתחת למכסה המנוע": החלפת שני ערכים ומיון בועות.

תזכורת

- `lst.sort()` ממין את הרשימה עצמה ומחזיר `None`. `sorted(lst)` מחזיר רשימה חדשה.
- מיון יורד: `reverse=True`. מחרוזות ממוינות לפי ABC, ואותיות גדולות באות לפני קטנות.
- החלפה: `a, b = b, a`. ברשימה: `lst[i], lst[i + 1] = lst[i + 1], lst[i]`.
- מיון בועות: משווים שכנים ומחליפים אם הם לא בסדר. אחרי כל מעבר הגדול ביותר "מבעבע" לסוף.

קל

מה יודפס? 1

```
a = [5, 2, 9, 1]
b = sorted(a)
print(a)
print(b)
a.sort(reverse=True)
print(a)
```

קל

המלכודת הקלאסית: מה יודפס? 2

```
lst = [3, 1, 2]
lst = lst.sort()
print(lst)
```

בינוני

מה יודפס? 3

```
names = ["dana", "Omer", "adam", "Lina"]
print(sorted(names))
```

בינוני

מה יודפס אחרי כל מעבר? 4

```
lst = [5, 2, 7, 1]
for p in range(len(lst) - 1):
    for i in range(len(lst) - 1 - p):
        if lst[i] > lst[i + 1]:
            lst[i], lst[i + 1] = lst[i + 1], lst[i]
    print(lst)
```

בינוני

מה יודפס? 5

```
grades = [72, 95, 88, 60, 95]
grades.sort(reverse=True)
print(grades[:3])
print(grades[len(grades) // 2])
```

6 מה הבאג? (המטרה: להדפיס את הרשימה ממוינת)

בינוני

```
lst = [4, 1, 3]
result = lst.sort()
print(result[0])
```

7 כמה **השוואות** עושה מיון בועות (הגרסה בשאלה 4) על רשימה של 5 איברים? ושל n איברים?

מאתגר

8 מה ההבדל בין `sorted(lst)[-1]` לבין `max(lst)`? איזה עדיף כשרוצים רק את הגדול ולמה?

בינוני

9 כתבו מיון בועות שמבצע מעבר נוסף **רק אם** במעבר הקודם הייתה לפחות החלפה אחת (משתנה `swapped`). הדפיסו כמה מעברים בוצעו.

בינוני

10 מיון בחירה: בכל שלב מוצאים את **הקטן ביותר** בחלק שעוד לא ממין ומחליפים אותו עם האיבר הראשון של החלק הזה. ממשו בלי `sort`.

מאתגר

11 קלטו 6 ציונים לרשימה. הדפיסו אותם ממוינים מהגבוה לנמוך, את שלושת הגבוהים, ואת הממוצע **בלי** הציון הגבוה והנמוך.

בינוני

11 פונקציות שאלות 13

הגדרה, פרמטרים, `return`, ערכי ברירת מחדל, משתנים מקומיים, ופונקציה שמקבלת רשימה.

תזכורת

- `def name(params)`: מגדירה. הקוד בפנים רץ רק כשקוראים לפונקציה.
- `return` מחזיר ערך **ויוצא** מהפונקציה. פונקציה בלי `return` מחזירה `None`.
- `print` מציג על המסך. `return` מחזיר ערך לקוד שקרא. אלה שני דברים שונים.
- פרמטר עם ברירת מחדל: `def f(a, b=2)`. אפשר לקרוא בשם: `f(b=1, a=4)`.
- משתנה שנוצר בתוך פונקציה הוא מקומי. רשימה שמועברת לפונקציה **יכולה** להשתנות בתוכה.

קל

מה יודפס?

1

```
def greet(name):  
    print("שלום", name)  
  
greet("דנה")  
greet("עומר")
```

בינוני

מה יודפס?

2

```
def f(a, b=2):  
    if a > b:  
        return a - b  
    return a * b  
  
print(f(5), f(1), f(3, 3), f(b=1, a=4))
```

בינוני

מה יודפס? `return` מול `print`

3

```
def show(x):  
    print(x * 2)  
  
def give(x):  
    return x * 2  
  
a = show(5)  
b = give(5)  
print(a, b)
```

```
def check(n):  
    for d in range(2, n):  
        if n % d == 0:  
            return False  
    return True  
  
print(check(7), check(9), check(2))
```

```
x = 10  
  
def change():  
    x = 99  
    print("inside:", x)  
  
change()  
print("outside:", x)
```

```
def add_bonus(grades):  
    for i in range(len(grades)):  
        grades[i] += 5  
  
my = [80, 90]  
add_bonus(my)  
print(my)
```

7 פונקציה שקוראת לפונקציה: מה יודפס?

בינוני

```
def square(x):  
    return x * x  
  
def sum_squares(a, b):  
    return square(a) + square(b)  
  
print(sum_squares(3, 4))  
print(square(sum_squares(1, 1)))
```

8 מה הבאג?

בינוני

```
def area(w, h):  
    result = w * h  
  
print(area(3, 4) + 1)
```

9 מה יודפס, ולמה?

בינוני

```
def f():  
    return 1  
    print("after")  
  
print(f())
```

10 כתבו פונקציה `digit_sum(n)` שמחזירה את סכום הספרות של `n`, ופונקציה `is_lucky(n)` שמחזירה `True` אם סכום הספרות מתחלק ב-7. הדפיסו את כל המספרים ה"מזליים" בין 1 ל-100.

קל

11 כתבו פונקציה `count_above(lst, limit)` שמחזירה כמה איברים ברשימה גדולים מ-`limit`. ברירת המחדל של `limit` היא 0.

קל

12 כתבו פונקציה `top3(lst)` שמחזירה רשימה של שלושת הערכים הגבוהים, מהגבוה לנמוך, בלי `sort` ו-`sorted`, ובלי לשנות את הרשימה המקורית. מה היא מחזירה כשיש פחות משלושה איברים?

מאתגר

13 תפריט: כתבו פונקציה `menu()` שמדפיסה אפשרויות וקולטת בחירה, ופונקציות `add(a, b)` ו- `power(a, b)`. הקוד הראשי ירוץ בלולאה עד שהמשתמש בוחר 0.

12 Tuple – טאפלים שאלות 12

רשימה שאי אפשר לשנות: יצירה, פריסה למשתנים, פונקציה שמחזירה כמה ערכים, ורשימה של tuple-ים כטבלה.

תזכורת

- `t = (3, 4)` tuple עם איבר אחד צריך פסיק: `(, 5)`.
- אינדקסים וחיתוך כמו ברשימה, אבל `t[0] = 9` זו שגיאה: tuple לא משתנה.
- רק שתי פעולות: `t.count(x)` ו-`t.index(x)`.
- פריסה: `a, b = t`. פונקציה יכולה להחזיר כמה ערכים: `return low, high`.
- רשימה של tuple-ים היא "טבלה": `for name, grade in students:`

קל

1 מה יודפס?

```
t = (4, 7, 4, 1)
a, b, c, d = t
print(t.count(4), t.index(7), a + d, sorted(t))
```

קל

2 מה יודפס?

```
t = (10, 20, 30, 40, 50)
print(t[1:4], t[-2], len(t), 30 in t)
```

בינוני

3 מה יודפס? מה הטיפוס של כל משתנה?

```
a = (5)
b = (5,)
print(type(a), type(b))
```

בינוני

מה יודפס?

4

```
def stats(lst):  
    return min(lst), max(lst), sum(lst) / len(lst)  
  
low, high, avg = stats([4, 8, 6])  
print(low, high, avg)  
print(stats([1, 2]))
```

בינוני

טבלה: מה יודפס?

5

```
students = [("dana", 88), ("omer", 54), ("lina", 97)]  
for name, grade in students:  
    if grade >= 55:  
        print(name)  
print(students[1][0], students[2][1])
```

בינוני

החלפה בעזרת tuple: מה יודפס?

6

```
a = 1  
b = 2  
c = 3  
a, b, c = c, a, b  
print(a, b, c)
```

קל

מה הבאג?

7

```
point = (3, 4)  
point[0] = 10  
print(point)
```

בינוני

מה הבאג?

8

```
a, b = (1, 2, 3)
```

10 כתבו פונקציה `min_max(lst)` שמחזירה tuple של (הקטן, הגדול) במעבר אחד על הרשימה, בלי `min` ו-`max`. קראו לה ופרסו את התוצאה לשני משתנים.

11 נתונה רשימה של (שם, ציון). הדפיסו את שם התלמיד עם הציון הגבוה, ואת רשימת השמות ממוינת לפי הציון מהגבוה לנמוך. רמז: הפכו כל זוג ל-(ציון, שם) ואז מיינו.

12 קלטו 3 נקודות (x ו-y לכל אחת) ושמרו אותן ברשימה של tuple-ים. הדפיסו את הנקודה הקרובה ביותר לראשית (0,0). אפשר להשוות את `x * x + y * y` במקום המרחק עצמו.

13 מילונים (Dictionaries)

מפתח ← ערך: יצירה, גישה, `in`, מעבר עם `keys()` ו-`items()`, ספירה במילון ומילון של רשימות. המילון מודגש בהנחיות לבחינה.

- `d = {key: value}` מילון ריק. `d[key] = value` מוסיף או מעדכן. `d[key]` קורא, ושגיאה (`KeyError`) אם המפתח לא קיים.
- `key in d` בודק אם מפתח קיים. `d.get(key, 0)` מחזיר 0 אם המפתח לא קיים, בלי שגיאה.
- `for k in d`: עובר על המפתחות. `for k, v in d.items()`: עובר על הזוגות.
- ספירה: `counts[w] = counts.get(w, 0) + 1`.
- מפתחות חייבים להיות ייחודיים. הצבה למפתח קיים דורסת את הערך הקודם.

```
ages = {"dana": 16, "omer": 17}
ages["lina"] = 15
ages["dana"] = 17
print(ages)
print(len(ages), "omer" in ages, 17 in ages)
```

```
prices = {"apple": 5, "bread": 8, "milk": 6}
total = 0
for item in prices:
    total += prices[item]
print(total)
for item, price in prices.items():
    if price > 5:
        print(item)
```

```
text = "to be or not to be"
counts = {}
for w in text.split():
    counts[w] = counts.get(w, 0) + 1
print(counts)
```

```
d = {"a": 1, "b": 2}
print(d.get("a"), d.get("z"), d.get("z", 0))
```

```
word = "mississippi"
counts = {}
for ch in word:
    if ch in counts:
        counts[ch] += 1
    else:
        counts[ch] = 1
best = ""
for ch in counts:
    if best == "" or counts[ch] > counts[best]:
        best = ch
print(counts)
print(best, counts[best])
```

```
words = ["apple", "avocado", "banana", "cherry", "blueberry"]
by_letter = {}
for w in words:
    first = w[0]
    if first not in by_letter:
        by_letter[first] = []
    by_letter[first].append(w)
print(by_letter)
print(len(by_letter["b"]))
```

```
stock = {"apple": 3, "milk": 0}
print(stock["bread"])
```

8 המטרה לספור מילים, אבל הקוד קורס. למה?

בינוני

```
counts = {}
for w in ["hi", "bye", "hi"]:
    counts[w] += 1
print(counts)
```

9 מה ההבדל בין רשימה למילון? מתי נבחר כל אחד? (תנו דוגמה)

בינוני

קל

10 נתון מילון של שם ← ציון. הדפיסו את הממוצע, את שם התלמיד עם הציון הגבוה, ואת כל מי שמעל הממוצע.

בינוני

11 ספר טלפונים: תפריט בלולאה: 1 הוספה, 2 חיפוש לפי שם, 3 הצגת כולם בסדר אלפביתי, 0 יציאה. חיפוש של שם שלא קיים מדפיס הודעה, בלי שגיאה.

מאתגר

12 נתונה רשימה של tuple-ים (שם, מקצוע, ציון). בנו מילון שם ← רשימת ציונים, והדפיסו לכל תלמיד את הממוצע שלו.

מאתגר

13 קלטו משפט והדפיסו כל מילה ומספר ההופעות שלה, מהשכיחה לנדירה. לא להבחין בין אותיות גדולות לקטנות (lower()).

14 עיבוד קבצים 12 שאלות

פתיחה, קריאה וכתיבה של קובצי טקסט, בשתי הצורות: open (...) with open ו- close כמו בבחינה. שורות ריקות, שורות הערה עם #, ו- split על כל שורה.

תזכורת

- f = open("a.txt", "r") ← עובדים f.close או f = open("a.txt") as f:with
- שסוגר לבד.
- מצבים: "r" קריאה, "w" כתיבה (מוחק את מה שהיה), "a" הוספה לסוף.
- for line in f: עובר שורה אחרי שורה. כל שורה מסתיימת ב-\n, ולכן עושים line.strip().
- f.read() כל הקובץ כמחרוזת אחת. f.readlines() או list(f) רשימה של שורות.
- f.write(text) לא מוסיף \n לבד. מספר צריך להמיר: f.write(str(n) + "\n")

```
with open("a.txt", "w") as f:
    f.write("one")
    f.write("two\n")
    f.write("three\n")

with open("a.txt") as f:
    lines = f.readlines()
print(lines)
print(len(lines))
```

```
f = open("log.txt", "w")
f.write("A\n")
f.close()
f = open("log.txt", "a")
f.write("B\n")
f.close()
f = open("log.txt", "w")
f.write("C\n")
f.close()
f = open("log.txt", "a")
f.write("D\n")
f.close()
f = open("log.txt", "r")
print(f.read())
f.close()
```

data.txt הקובץ

5
12

7

```
total = 0
count = 0
with open("data.txt") as f:
    for line in f:
        line = line.strip()
        if line != "":
            total += int(line)
            count += 1
print(total, count, total / count)
```

grades.txt הקובץ

name grade
dana 88
omer 54
lina 97

adam 71

```
f = open("grades.txt", "r")
lines = list(f)
f.close()
for line in lines:
    if line[0] != "#" and line.strip() != "":
        parts = line.split()
        if int(parts[1]) >= 70:
            print(parts[0])
```

poem.txt הקובץ

```
the sun is up
the day is long
up up and away
```

```
with open("poem.txt") as f:
    text = f.read()
    print(len(text.split("\n")) - 1, len(text.split()), text.count("up"))
```

```
f = open("nums.txt", "w")
for i in range(3):
    f.write(i)
f.close()
```

```
with open("scores.txt") as f:
    for line in f:
        print(line)
```

8 למה כשמדפיסים שורות מקובץ עם `print(line)` מופיעה שורה ריקה בין כל שתי שורות? איך מתקנים?

9 כתבו לקובץ `squares.txt` את המספרים 1 עד 10 ואת הריבוע של כל אחד, שורה לכל מספר (3 9). אחר כך קראו את הקובץ והדפיסו את סכום הריבועים.

10 הקובץ `grades.txt` (מוצג למטה) מכיל שם וציון בכל שורה, שורות הערה שמתחילות ב-`#` ושורות ריקות. הדפיסו את הממוצע ואת שם בעל הציון הגבוה.

grades.txt הקובץ

```
# name grade
dana 88
omer 54
lina 97

adam 71
```

11 העתיקו מ-`grades.txt` לקובץ חדש `passed.txt` רק את השורות של מי שעבר (55 ומעלה), ובסוף הוסיפו שורה עם מספר העוברים.

grades.txt הקובץ

```
# name grade
dana 88
omer 54
lina 97

adam 71
```

12 "תוכנית שזוכרת": קראו מהקובץ `counter.txt` מספר, הוסיפו לו 1, הדפיסו אותו ושמרו בחזרה. אם הקובץ לא קיים, התחילו מ-0 (`try` / `except FileNotFoundError`). הריצו את התוכנית כמה פעמים.

15 עבודה מתקדמת עם קבצים

שאלות 12

מחברים הכול: חיפוש בקובץ, ניתוח טקסט עם מילון, העתקה עם שינוי, וקובץ CSV שנטען לרשימה של tuple-ים או למילון.

תזכורת

- CSV: כל שורה היא רשומה, והשדות מופרדים בפסיק: `name, grade = line.strip().split(",")`.
- השורה הראשונה היא לרוב כותרות. מדלגים עליה עם `f.readline()` לפני הלולאה.
- טוענים את הקובץ פעם אחת לזיכרון (רשימה או מילון), ומחשבים עליו.
- `word in line.split()` בודק מילה שלמה. `word in line` מוצא גם חלק ממילה ("up" בתוך "cup").

- מספר שורה: מונה `line_no += 1` בכל סיבוב של הלולאה.
- אל תפתחו לכתיבה את אותו קובץ שאתם קוראים: "w" מוחק אותו מיד.

קל

מה יודפס?

1

poem.txt הקובץ

```
the sun is up
the day is long
up up and away
```

```
word = "up"
line_no = 0
with open("poem.txt") as f:
    for line in f:
        line_no += 1
        if word in line.split():
            print(line_no, line.strip())
```

קל

מה יודפס?

2

scores.csv הקובץ

```
name,class,grade
dana,a1,88
omer,a2,54
lina,a1,97
adam,a2,71
noa,a1,65
```

```
with open("scores.csv") as f:
    header = f.readline()
    rows = []
    for line in f:
        name, cls, grade = line.strip().split(",")
        rows.append((name, cls, int(grade)))
print(header.strip())
print(len(rows), rows[0])
```

הקובץ scores.csv

```
name,class,grade
dana,a1,88
omer,a2,54
lina,a1,97
adam,a2,71
noa,a1,65
```

```
totals = {}
counts = {}
with open("scores.csv") as f:
    f.readline()
    for line in f:
        name, cls, grade = line.strip().split(",")
        totals[cls] = totals.get(cls, 0) + int(grade)
        counts[cls] = counts.get(cls, 0) + 1
for cls in sorted(totals):
    print(cls, totals[cls] / counts[cls])
```

הקובץ poem.txt

```
the sun is up
the day is long
up up and away
```

```
counts = {}
with open("poem.txt") as f:
    for line in f:
        for w in line.split():
            counts[w] = counts.get(w, 0) + 1
best = max(counts, key=counts.get)
print(best, counts[best])
```

הקובץ poem.txt

```
the sun is up
the day is long
up up and away
```

```
with open("poem.txt") as src:
    text = src.read()
with open("poem2.txt", "w") as dst:
    dst.write(text.replace("up", "UP"))
print(text.count("up"), "replacements")
```

6 השורה האחרונה אמורה להמיר את הציון, אבל התוכנית קורסת כבר בשורה הראשונה של הנתונים. למה?

הקובץ scores.csv

```
name,class,grade
dana,a1,88
omer,a2,54
lina,a1,97
adam,a2,71
noa,a1,65
```

```
total = 0
with open("scores.csv") as f:
    for line in f:
        name, cls, grade = line.strip().split(",")
        total += int(grade)
print(total)
```

7 קוד שקורא את notes.txt ובאותו זמן פותח אותו עם open("notes.txt", "w") כדי לכתוב גרסה מתוקנת מאבד את כל התוכן. למה? מה הדרך הנכונה?

8 כתבו פונקציה find_word(filename, word) שמחזירה רשימה של מספרי השורות שבהן המילה מופיעה. אם המילה לא נמצאה, הדפיסו הודעה מתאימה.

בינוני

9 לקובץ `poem.txt` הדפיסו: מספר שורות, מספר מילים, מספר תווים (בלי `\n`), והשורה הארוכה ביותר.

בינוני

10 מ- `scores.csv` (שם, כיתה, ציון) כתבו קובץ `report.txt` : שורה לכל תלמיד בפורמט (עבר/נכשל) `name: grade` , ממיון מהציון הגבוה לנמוך, ובסוף שורת ממוצע.

הקובץ `scores.csv`

```
name,class,grade
dana,a1,88
omer,a2,54
lina,a1,97
adam,a2,71
noa,a1,65
```

מתגר

11 אינדקס מילים: קראו את `poem.txt` ובנו מילון מילה ← רשימת מספרי השורות שבהן היא מופיעה (בלי כפילויות, בלי להבחין בין אותיות גדולות לקטנות). הדפיסו את המילים בסדר אלפביתי.

מתגר

12 פרויקט קטן: מלאי חנות. טענו את `stock.txt` (שורה: `item,amount`) למילון. קלטו מכירות (שם מוצר וכמות) עד שמוקלד `end` , עדכנו את המלאי (לא לרדת מתחת ל-0) ושמרו את המילון בחזרה לקובץ.

הקובץ `stock.txt`

```
milk,5
bread,3
eggs,12
```

מפתח תשובות

נסו לפתור לבד לפני שאתם פותחים. במשימות "כתבו תוכנית" יש יותר מפתרון נכון אחד. הפתרון כאן הוא דוגמה, ולידו ההרצה שלו (הקלט שהוקלד מופיע אחרי ההודעה).

פלט

1.

2 4 6 8 10

פלט

2.

10
7
4
1

פלט

3.

12 10

מעקב: $i = 1, 4, 7$ נכנסים ללולאה, וכש- $i = 10$ התנאי False. $total = 1 + 4 + 7$.

פלט

4.

3

פלט

5.

12

הספרות נשלפות מהסוף להתחלה: 1, 7, 0, 4.

פלט

6.

1 2 4 5 7

3 ו-6 מדולגים בגלל continue, וב-8 הלולאה נעצרת לפני ההדפסה.

פלט

7.

1

22
333

פלט

8.

מספר: 5
מספר: 8
מספר: 2
מספר: 0
15 3

ה-0 הוא הזקיף: הוא עוצר את הלולאה ולא נספר.

9. `RuntimeError: infinite loop`

השורה `i += 1` לא מוזחת, ולכן היא **מחוץ** ללולאה. `i` נשאר 1 והתנאי תמיד `True`. התיקון: להזיח את `i += 1` לתוך גוף ה-`while`.

10. (א) 3 פעמים: 3, 7, 11. (ב) 0 פעמים: ה-`range` ריק כי `start = stop` (ג) 0 פעמים: התנאי `False` כבר בבדיקה הראשונה.

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
n = int(input("מספר: "))
digits = 0
total = 0
evens = 0
while n > 0:
    d = n % 10
    digits += 1
    total += d
    if d % 2 == 0:
        evens += 1
    n //= 10
print(digits, total, evens)
```

הרצה לדוגמה

מספר: 1234
4 10 2

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```
n = int(input("מספר: "))
rev = 0
while n > 0:
    d = n % 10
    rev = rev * 10 + d
    n //= 10
print(rev)
```

הרצה לדוגמה

מספר: 3720
273

ה-0 שהיה בסוף הופך לאפס מוביל, ולכן הוא נעלם.

13. פתרון לדוגמה (יש עוד דרכים נכונות):

```
count = 0
total = 0
fails = 0
best = 0
grade = int(input("ציון: "))
while grade != -1:
    count += 1
    total += grade
    if grade < 55:
        fails += 1
    if grade > best:
        best = grade
    grade = int(input("ציון: "))
if count == 0:
    print("אין ציונים")
else:
    print("ממוצע:", total / count)
    print("הכי גבוה:", best)
    print("נכשלים:", fails)
```

הרצה לדוגמה

ציון: 80
ציון: 45
ציון: 92
ציון: 50
ציון: 1-
ממוצע: 66.75
הכי גבוה: 92
נכשלים: 2

14. פתרון לדוגמה (יש עוד דרכים נכונות):

```
for i in range(1, 6):  
    for j in range(1, 6):  
        print(i * j, end="\t")  
    print()
```

הרצה לדוגמה

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

15. פתרון לדוגמה (יש עוד דרכים נכונות):

```
for n in range(2, 1001):  
    total = 0  
    for d in range(1, n):  
        if n % d == 0:  
            total += d  
    if total == n:  
        print(n)
```

הרצה לדוגמה

6
28
496

פלט

1.

True
True
False

פלט

2.

True
False

בשורה הראשונה `and` מחושב קודם: `b and not c` הוא False, ואז `True or False` הוא True.

פלט

3.

False True

פלט

4.

0b1100 0b1010
8 14 6

$0110 = 1010 \wedge 1100$ (6), $1110 = 1010 \vee 1100$ (14), $1000 = 1010 \& 1100$ (8).

פלט

5.

10 40 10 3

הזזה שמאלה ב-k מכפילה ב-2 בחזקת k, והזזה ימינה מחלקת (חלוקה שלמה).

פלט

6.

False
True

בשתי השורות Python יודעת את התשובה כבר מהצד השמאלי, ולכן לא מחשבת את `10 / x`.

7. (א) `age >= 13 and age <= 19` או `13 <= age <= 19` (ב).
(א) `ch == "a" or ch == "e" or ch == "i" or ch == "o" or ch == "u"`
(ג) `(year % 4 == 0 and year % 100 != 0) or year % 400 == 0` (ח) `ch in "aeiou"`

8. `AssertionError: the condition is always True`

הביטוי מתפרש כ- `(grade == 90) or 100`. 100 הוא ערך "אמיתי" (כל מספר שאינו 0), ולכן התנאי תמיד True.
התיקון: `grade == 90 or grade == 100`.

9. (א) `0b0111`: הדלקת ביט 1. (ב) `0b0100`: בדיקה, ביט 2 דולק. (ג) `0b0100`: היפוך ביט 0, שהיה דולק ונכבה.

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```
a = int(input("צלע א: "))
b = int(input("צלע ב: "))
c = int(input("צלע ג: "))
if not (a + b > c and a + c > b and b + c > a):
    print("מחולש")
elif a == b and b == c:
    print("שווה צלעות")
elif a == b or b == c or a == c:
    print("שווה שוקיים")
else:
    print("שונה צלעות")
```

הרצה לדוגמה

```
צלע א: 5
צלע ב: 5
צלע ג: 8
שווה שוקיים
```

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
n = int(input("מספר: "))
if n & 1 == 0:
    print("זוגי")
```

```
else:  
    print("אי-זוגי")
```

הרצה לדוגמה

מספר: 14
זוגי

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```
n = int(input("מספר: "))  
ones = 0  
while n > 0:  
    ones += n & 1  
    n >>= 1  
print(ones)
```

הרצה לדוגמה

מספר: 13
3

פלט

1.

```
3 4 [8, 1] [3, 8] 5
```

פלט

2.

```
[8, 7, 1, 9, 4, 12]
```

פלט

3.

```
2 1 False
['dog', 'cat', 'bird']
```

`remove` מוחק רק את המופע הראשון.

פלט

4.

```
[1, 2, 3, 4]
[1, 2, 3]
```

`b = a` לא מעתיק: a ו-b הם אותה רשימה. `a[:]` יוצר עותק חדש.

פלט

5.

```
3
```

הזוגות השווים: (4,4) במקומות 0-1, ו-(2,2) פעמיים: 3-4 ו-4-5.

פלט

6.

```
[1, 2, 0, 3, 0, 0, 0, 0, 0, 1]
3
```

השורה השנייה מדפיסה את הספרה שהופיעה הכי הרבה פעמים: המיקום של הערך הגדול ברשימת המונים.

```
[5, 0, 0, 1, 9]
```

.8 `IndexError: list index out of range`

ברשימה של 3 איברים האינדקסים הם 0,1,2. `range(len(lst) + 1)` מגיע עד 3. התיקון:
`. range(len(lst))`

.9 בצורה הראשונה מקבלים את הערכים עצמם. בשנייה מקבלים את המיקומים. צריך את המיקום כשמשנים איבר ברשימה
`( lst[i] = ... )` או כשמשווים לאיבר שלידו `( lst[i + 1] )`.

.10 פתרון לדוגמה (יש עוד דרכים נכונות):

```
nums = []
x = int(input("מספר: "))
while x != 0:
    nums.append(x)
    x = int(input("מספר: "))
if len(nums) > 0:
    avg = sum(nums) / len(nums)
    print("ממוצע:", avg)
    for n in nums:
        if n > avg:
            print(n)
```

הרצה לדוגמה

```
מספר: 4
מספר: 10
מספר: 7
מספר: 1
מספר: 0
ממוצע: 5.5
10
7
```

.11 פתרון לדוגמה (יש עוד דרכים נכונות):

```

lst = [12, 45, 7, 45, 30]
best = 0
for i in range(1, len(lst)):
    if lst[i] > lst[best]:
        best = i
print("הגדול:", lst[best], "במקום", best)

```

הרצה לדוגמה

הגדול: 45 במקום 1

השוואה עם > (ולא >=) שומרת את המופע הראשון של הגדול.

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```

lst = [1, 3, 5, 2, 4, 6, 8, 1]
best = 1
current = 1
for i in range(len(lst) - 1):
    if lst[i + 1] > lst[i]:
        current += 1
        if current > best:
            best = current
    else:
        current = 1
print(best)

```

הרצה לדוגמה

4

13. פתרון לדוגמה (יש עוד דרכים נכונות):

```

import random
random.seed(1)
counts = [0] * 7 # place 0 is not used
for _ in range(600):
    counts[random.randint(1, 6)] += 1
for face in range(1, 7):
    print(face, counts[face])

```

הרצה לדוגמה

1 104

2 102

3 103

4 94

5 113

6 84

כל מספר יוצא בערך 100 פעמים. בלי **seed** התוצאה שונה בכל הרצה.

פלט

1.

```
[5, 2, 9, 1]
[1, 2, 5, 9]
[9, 5, 2, 1]
```

פלט

2.

None

`sort` משנה את הרשימה ומחזיר `None`, וזה מה שנשמר ב- `lst`.

פלט

3.

```
['Lina', 'Omer', 'adam', 'dana']
```

אותיות גדולות באות לפני קטנות בסדר של Python.

פלט

4.

```
[2, 5, 1, 7]
[2, 1, 5, 7]
[1, 2, 5, 7]
```

אחרי המעבר הראשון 7 בסוף, אחרי השני גם 5 במקומו.

פלט

5.

```
[95, 95, 88]
88
```

השורה השנייה היא החציון: האיבר האמצעי ברשימה ממוינת.

6. `TypeError: 'NoneType' object is not subscriptable`

`lst.sort()` מחזיר `None`, ואי אפשר לגשת ל- `None[0]`. התיקון: `result = sorted(lst)`, או `lst.sort()` ואז `print(lst[0])`.

7. $10 = 1 + 2 + 3 + 4$. באופן כללי $1 + 2 + \dots + (n-1) + n = n(n-1)/2$. מספר ההשוואות לא תלוי בסדר ההתחלתי.

8. שניהם נותנים את הגדול. `max` עובר על הרשימה פעם אחת, ו- `sorted` ממין את כולה ובונה רשימה חדשה. לכן `max` עדיף.

9. פתרון לדוגמה (יש עוד דרכים נכונות):

```
lst = [1, 2, 4, 3, 5]
passes = 0
swapped = True
while swapped:
    swapped = False
    passes += 1
    for i in range(len(lst) - 1):
        if lst[i] > lst[i + 1]:
            lst[i], lst[i + 1] = lst[i + 1], lst[i]
            swapped = True
print(lst, "מעברים:", passes)
```

הרצה לדוגמה

[5, 4, 3, 2, 1] מעברים: 2

ברשימה כמעט ממוינת מספיקים 2 מעברים: אחד שמתקן, ואחד שמוודא שאין מה להחליף.

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```
lst = [29, 10, 14, 37, 13]
for i in range(len(lst) - 1):
    small = i
    for j in range(i + 1, len(lst)):
        if lst[j] < lst[small]:
            small = j
```

```
lst[i], lst[small] = lst[small], lst[i]
print(lst)
```

הרצה לדוגמה

```
[10, 29, 14, 37, 13]
[10, 13, 14, 37, 29]
[10, 13, 14, 37, 29]
[10, 13, 14, 29, 37]
```

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
grades = []
for i in range(6):
    grades.append(int(input("ציון: ")))
grades.sort(reverse=True)
print(grades)
print("שלושת הגבוהים:", grades[:3])
middle = grades[1:-1]
print("ממוצע בלי קצוות:", sum(middle) / len(middle))
```

הרצה לדוגמה

```
ציון: 70
ציון: 100
ציון: 85
ציון: 40
ציון: 90
ציון: 75
[100, 90, 85, 75, 70, 40]
שלושת הגבוהים: [85, 90, 100]
ממוצע בלי קצוות: 80.0
```

פלט

1.

שלום דנה
שלום עומר

פלט

2.

3 2 9 3

פלט

3.

10
None 10

show מדפיסה 10 בעצמה, אבל לא מחזירה כלום. לכן **a** הוא **None**.

פלט

4.

True False True

return בתוך הלולאה יוצא מהפונקציה מיד. ל-2 ה-range ריק, ולכן מוחזר True.

פלט

5.

inside: 99
outside: 10

פלט

6.

[85, 95]

הפונקציה קיבלה את **אותה** רשימה, ולכן השינוי נשאר גם אחרי שהיא מסתיימת.

פלט

7.

25

8. `TypeError: unsupported operand type(s) for +: 'NoneType' and 'int'`

הפונקציה מחשבת אבל לא מחזירה, ולכן `area(3, 4)` הוא `None`, ו-`None + 1` זו שגיאה. התיקון: להוסיף `. return result`

9. יודפס `1` בלבד. `return` יוצא מהפונקציה, ולכן השורה `print("after")` לא רצה אף פעם ("קוד מת").

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```
def digit_sum(n):
    total = 0
    while n > 0:
        total += n % 10
        n //= 10
    return total

def is_lucky(n):
    return digit_sum(n) % 7 == 0

for n in range(1, 101):
    if is_lucky(n):
        print(n, end=" ")
```

הרצה לדוגמה

7 16 25 34 43 52 59 61 68 70 77 86 95

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
def count_above(lst, limit=0):
    count = 0
    for x in lst:
        if x > limit:
            count += 1
    return count
```

```
print(count_above([3, -1, 8, 0, 5]))
print(count_above([3, -1, 8, 0, 5], 4))
```

הרצה לדוגמה

```
3
2
```

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```
def top3(lst):
    work = lst[:]          # a copy - the caller's list stays the same
    result = []
    while len(work) > 0 and len(result) < 3:
        best = work[0]
        for x in work:
            if x > best:
                best = x
        result.append(best)
        work.remove(best)
    return result

nums = [5, 9, 2, 7, 9]
print(top3(nums))
print(nums)
print(top3([4]))
```

הרצה לדוגמה

```
[9, 9, 7]
[5, 9, 2, 7, 9]
[4]
```

כשיש פחות משלושה איברים הפונקציה מחזירה את מה שיש.

13. פתרון לדוגמה (יש עוד דרכים נכונות):

```
def menu():
    print("חיבור (2 חזקה 0) יציאה (1)")
    return int(input("בחירה: "))
```

```

def add(a, b):
    return a + b

def power(a, b):
    result = 1
    for _ in range(b):
        result *= a
    return result

choice = menu()
while choice != 0:
    a = int(input("a: "))
    b = int(input("b: "))
    if choice == 1:
        print(add(a, b))
    elif choice == 2:
        print(power(a, b))
    choice = menu()

```

הרצה לדוגמה

```

(1 חיבור (2 חזקה (0 יציאה
בחירה: 1
a: 3
b: 4
7
(1 חיבור (2 חזקה (0 יציאה
בחירה: 2
a: 2
b: 10
1024
(1 חיבור (2 חזקה (0 יציאה
בחירה: 0

```

פלט

1.

```
2 1 5 [1, 4, 4, 7]
```

`sorted` על tuple מחזירה **רשימה**.

פלט

2.

```
(20, 30, 40) 40 5 True
```

פלט

3.

```
<class 'int'> <class 'tuple'>
```

סוגריים לבד הם רק סוגריים. מה שיוצר tuple הוא **הפסיק**.

פלט

4.

```
4 8 6.0
(1, 2, 1.5)
```

פלט

5.

```
dana
lina
omer 97
```

פלט

6.

```
3 1 2
```

הצד הימני מחושב **כולו** לפני ההצבה: (2,1,3).

7. `TypeError: 'tuple' object does not support item assignment`

tuple לא ניתן לשינוי. אפשר ליצור tuple חדש: `point = (10, point[1])`, או להשתמש ברשימה.

8. `ValueError: too many values to unpack (expected 2)`

בפריסה מספר המשתנים חייב להיות שווה למספר האיברים: כאן 2 משתנים ו-3 איברים. התיקון:

```
a, b, c = (1, 2, 3)
```

9. tuple: נתונים קבועים שלא אמורים להשתנות, ובדרך כלל מסוגים שונים, כמו נקודה `(x, y)` או רשומה

.רשימה: אוסף שגדל, קטן או משתנה, כמו רשימת ציונים שמוסיפים לה. `("dana", 88)`

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```
def min_max(lst):
    low = lst[0]
    high = lst[0]
    for x in lst:
        if x < low:
            low = x
        if x > high:
            high = x
    return low, high

lo, hi = min_max([7, 3, 9, 1, 5])
print(lo, hi)
```

הרצה לדוגמה

1 9

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
students = [("dana", 88), ("omer", 54), ("lina", 97), ("adam", 71)]
flipped = []
for name, grade in students:
    flipped.append((grade, name))
flipped.sort(reverse=True)
print("הכי גבוה:", flipped[0][1])
for grade, name in flipped:
    print(name, grade)
```

הרצה לדוגמה

```
lina: הכי גבוה:
lina 97
dana 88
adam 71
omer 54
```

tuple-ים ממוינים לפי האיבר הראשון, ולכן הציון צריך להיות ראשון.

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```
points = []
for i in range(3):
    x = int(input("x: "))
    y = int(input("y: "))
    points.append((x, y))
best = points[0]
for x, y in points:
    if x * x + y * y < best[0] ** 2 + best[1] ** 2:
        best = (x, y)
print("הכי קרובה:", best)
```

הרצה לדוגמה

```
x: 3
y: 4
x: -1
y: 2
x: 5
y: 0
הכי קרובה: (-1, 2)
```

פלט

1.

```
{'dana': 17, 'omer': 17, 'lina': 15}
3 True False
```

`in` בודק מפתחות, ולכן `17 in ages` הוא `False`, אף ש-17 הוא ערך במילון.

פלט

2.

```
19
bread
milk
```

פלט

3.

```
{'to': 2, 'be': 2, 'or': 1, 'not': 1}
```

פלט

4.

```
1 None 0
```

פלט

5.

```
{'m': 1, 'i': 4, 's': 4, 'p': 2}
i 4
```

i ו-s שתייהן 4. > שומר את מי שהגיעה ראשונה במילון (i).

פלט

6.

```
{'a': ['apple', 'avocado'], 'b': ['banana', 'blueberry'], 'c': ['cherry']}
2
```

7. `KeyError: 'bread'`

אין במילון מפתח "bread". בודקים קודם עם `if "bread" in stock:`, או משתמשים ב-
`. stock.get("bread", 0)`

8. `KeyError: 'hi'`

בפעם הראשונה ש- "hi" מגיעה, אין לה עדיין ערך, ולכן `counts[w] += 1` נכשל. התיקון:
`. counts[w] = counts.get(w, 0) + 1`

9. ברשימה ניגשים לפי **מיקום** (0,1,2,...), ובמילון לפי **מפתח** משמעותי. רשימה מתאימה לרצף של ערכים, כמו ציונים לפי הסדר. מילון מתאים לחיפוש לפי שם, כמו ספר טלפונים: שם ← מספר.

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```
grades = {"dana": 88, "omer": 54, "lina": 97, "adam": 71}
avg = sum(grades.values()) / len(grades)
print("ממוצע:", avg)
best = ""
for name in grades:
    if best == "" or grades[name] > grades[best]:
        best = name
print("הכי גבוה:", best)
for name, g in grades.items():
    if g > avg:
        print(name)
```

הרצה לדוגמה

```
ממוצע: 77.5
הכי גבוה: lina
dana
lina
```

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
book = {}
choice = input("(1/2/3/0) בחירה: ")
while choice != "0":
    if choice == "1":
        name = input("שם: ")
```

```

        book[name] = input("טלפון: ")
    elif choice == "2":
        name = input("שם: ")
        if name in book:
            print(book[name])
        else:
            print("לא נמצא")
    elif choice == "3":
        for name in sorted(book):
            print(name, book[name])
    choice = input("(1/2/3/0) בחירה: ")

```

הרצה לדוגמה

```

1 : (1/2/3/0) בחירה
שם: omer
טלפון: 050-1111111
1 : (1/2/3/0) בחירה
שם: dana
טלפון: 052-2222222
2 : (1/2/3/0) בחירה
שם: lina
לא נמצא
3 : (1/2/3/0) בחירה
dana 052-2222222
omer 050-1111111
0 : (1/2/3/0) בחירה

```

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```

records = [("dana", "math", 90), ("omer", "math", 60), ("dana", "english",
    ("omer", "english", 75), ("lina", "math", 100)]
by_student = {}
for name, subject, grade in records:
    if name not in by_student:
        by_student[name] = []
    by_student[name].append(grade)
for name, grades in by_student.items():
    print(name, sum(grades) / len(grades))

```

הרצה לדוגמה

```
dana 85.0
omer 67.5
lina 100.0
```

13. פתרון לדוגמה (יש עוד דרכים נכונות):

```
sentence = input("משפט: ")
counts = {}
for w in sentence.lower().split():
    counts[w] = counts.get(w, 0) + 1
pairs = []
for w, c in counts.items():
    pairs.append((c, w))
pairs.sort(reverse=True)
for c, w in pairs:
    print(w, c)
```

הרצה לדוגמה

```
The cat and the hat and THE bat :משפט
the 3
and 2
hat 1
cat 1
bat 1
```

מיון של (מספר, מילה) מהגדול לקטן. כשהמספרים שווים, המילים ממוינות בסדר הפוך (z לפני a).

פלט

1.

```
['onetwo\n', 'three\n']
2
```

`write` לא מוסיף ירידת שורה, ולכן one ו-two באותה שורה.

a.txt הקובץ

```
onetwo
three
```

פלט

2.

```
C
D
```

log.txt הקובץ

```
C
D
```

פלט

3.

```
24 3 8.0
```

בלי הבדיקה, `int("")` היה זורק `ValueError` על השורה הריקה.

פלט

4.

```
dana
lina
adam
```

שורה שמתחילה ב- `#` היא הערה, ומדלגים עליה. `split()` מחלק לפי רווחים.

פלט

5.

המספר הראשון הוא מספר השורות (הקובץ מסתיים ב- `\n` , ולכן יש חלק ריק בסוף), השני מספר המילים, והשלישי כמה פעמים מופיע הרצף "up".

6. `TypeError: write() argument must be str, not int`

`write` מקבל רק מחרוזת. התיקון: `f.write(str(i) + "\n")` , כי צריך גם ירידת שורה.

7. `FileNotFoundError: [Errno 2] No such file or directory: 'scores.txt'`

אין קובץ בשם `scores.txt` בתיקייה שבה התוכנית רצה. בודקים את שם הקובץ ואת המיקום, או תופסים את השגיאה עם `except FileNotFoundError / try`.

8. כל שורה בקובץ כבר מסתיימת ב- `\n` , ו- `print` מוסיף עוד אחת. מתקנים עם `print(line.strip())` (או `print(line, end="")`).

9. פתרון לדוגמה (יש עוד דרכים נכונות):

```
f = open("squares.txt", "w")
for n in range(1, 11):
    f.write(str(n) + " " + str(n * n) + "\n")
f.close()

total = 0
f = open("squares.txt", "r")
for line in f:
    parts = line.split()
    total += int(parts[1])
f.close()
print(total)
```

הרצה לדוגמה

385

squares.txt הקובץ

1 1

```
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
```

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```
count = 0
total = 0
best_name = ""
best = -1
f = open("grades.txt", "r")
for line in f:
    line = line.strip()
    if line == "" or line[0] == "#":
        continue
    parts = line.split()
    grade = int(parts[1])
    total += grade
    count += 1
    if grade > best:
        best = grade
        best_name = parts[0]
f.close()
print("ממוצע:", total / count)
print("הכי גבוה:", best_name)
```

הרצה לדוגמה

```
ממוצע: 77.5
הכי גבוה: lina
```

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
src = open("grades.txt", "r")
dst = open("passed.txt", "w")
```

```

passed = 0
for line in src:
    parts = line.split()
    if len(parts) == 2 and parts[0][0] != "#" and int(parts[1]) >= 55:
        dst.write(line)
        passed += 1
dst.write("passed: " + str(passed) + "\n")
src.close()
dst.close()
print("נכתבו", passed, "שורות")

```

הרצה לדוגמה

נכתבו 3 שורות

הקובץ passed.txt

```

dana 88
lina 97
adam 71
passed: 3

```

מדלג גם על השורה הריקה (שם `parts` ריק). `len(parts) == 2`

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```

try:
    f = open("counter.txt", "r")
    count = int(f.read())
    f.close()
except FileNotFoundError:
    count = 0
count += 1
print("התוכנית רצה", count, "פעמים")
f = open("counter.txt", "w")
f.write(str(count))
f.close()

```

הקובץ counter.txt

4

הרצה לדוגמה

התוכנית רצה 5 פעמים

בהרצה לדוגמה הקובץ כבר הכיל 4, כאילו התוכנית רצה 4 פעמים קודם.

פלט

1.

```
1 the sun is up
3 up up and away
```

פלט

2.

```
name,class,grade
5 ('dana', 'a1', 88)
```

פלט

3.

```
a1 83.33333333333333
a2 62.5
```

פלט

4.

```
up 3
```

`max(d, key=d.get)` מחזיר את המפתח שהערך שלו הכי גדול.

פלט

5.

```
3 replacements
```

הקובץ poem2.txt

```
the sun is UP
the day is long
UP UP and away
```

6. `ValueError: invalid literal for int() with base 10: 'grade'`

השורה הראשונה היא שורת הכותרות, ו-`int("grade")` נכשל. התיקון: `f.readline()` לפני הלולאה, כדי לדלג על הכותרות.

7. פתיחה במצב "w" מרוקנת את הקובץ מיד, עוד לפני שקראנו ממנו. הדרך הנכונה: לקרוא קודם את כל התוכן לזיכרון ולסגור, ורק אז לפתוח לכתיבה. אפשרות אחרת: לכתוב לקובץ חדש.

8. פתרון לדוגמה (יש עוד דרכים נכונות):

```
def find_word(filename, word):
    result = []
    line_no = 0
    f = open(filename, "r")
    for line in f:
        line_no += 1
        if word in line.split():
            result.append(line_no)
    f.close()
    return result

for w in ["is", "sky"]:
    lines = find_word("poem.txt", w)
    if len(lines) > 0:
        print(w, lines)
    else:
        print(w, "לא נמצאה")
```

הקובץ poem.txt

```
the sun is up
the day is long
up up and away
```

הרצת הדוגמה

```
is [1, 2]
sky לא נמצאה
```

9. פתרון לדוגמה (יש עוד דרכים נכונות):

```
lines = 0
words = 0
chars = 0
longest = ""
```

```

with open("poem.txt") as f:
    for line in f:
        line = line.strip()
        lines += 1
        words += len(line.split())
        chars += len(line)
        if len(line) > len(longest):
            longest = line
print(lines, words, chars)
print("הארוכה:", longest)

```

הקובץ poem.txt

```

the sun is up
the day is long
up up and away

```

הרצה לדוגמה

```

3 12 42
the day is long :הארוכה

```

10. פתרון לדוגמה (יש עוד דרכים נכונות):

```

rows = []
with open("scores.csv") as f:
    f.readline()
    for line in f:
        name, cls, grade = line.strip().split(",")
        rows.append((int(grade), name))
rows.sort(reverse=True)
total = 0
with open("report.txt", "w") as out:
    for grade, name in rows:
        if grade >= 55:
            status = "עבר"
        else:
            status = "נכשל"
        out.write(f"{name}: {grade} ({status})\n")
        total += grade
    out.write(f"ממוצע: {total / len(rows)}\n")
print("נכתב report.txt")

```

הרצה לדוגמה

נכתב report.txt

report.txt הקובץ

lina: 97 (עבר)
dana: 88 (עבר)
adam: 71 (עבר)
noa: 65 (עבר)
omer: 54 (נכשל)
ממוצע: 75.0

11. פתרון לדוגמה (יש עוד דרכים נכונות):

```
index = {}  
line_no = 0  
with open("poem.txt") as f:  
    for line in f:  
        line_no += 1  
        for w in line.lower().split():  
            if w not in index:  
                index[w] = []  
            if line_no not in index[w]:  
                index[w].append(line_no)  
for w in sorted(index):  
    print(w, index[w])
```

poem.txt הקובץ

the sun is up
the day is long
up up and away

הרצה לדוגמה

and [3]
away [3]
day [2]
is [1, 2]
long [2]
sun [1]

```
the [1, 2]
up [1, 3]
```

"up" מופיעה פעמיים בשורה 3, אבל מספר השורה נשמר פעם אחת.

12. פתרון לדוגמה (יש עוד דרכים נכונות):

```
stock = {}
with open("stock.txt") as f:
    for line in f:
        item, amount = line.strip().split(",")
        stock[item] = int(amount)

item = input("מוצר: ")
while item != "end":
    qty = int(input("כמות: "))
    if item not in stock:
        print("אין מוצר כזה")
    elif qty > stock[item]:
        print("אין מספיק במלאי, נשאר", stock[item])
    else:
        stock[item] -= qty
    item = input("מוצר: ")

with open("stock.txt", "w") as f:
    for item, amount in stock.items():
        f.write(item + "," + str(amount) + "\n")
print(stock)
```

הרצה לדוגמה

```
מוצר: milk
כמות: 2
מוצר: bread
כמות: 9
אין מספיק במלאי, נשאר 3
מוצר: eggs
כמות: 1
אין מוצר כזה
מוצר: end
{'milk': 3, 'bread': 3, 'eggs': 12}
```

מכינה תקשוב · פייתון — תרגול נלווה ליחידות 7-15 · שאלון 735911