

מטלת מנהה (ממ"ז) 11

הקורס: תוכנות וניתוח נתוניים בשפה פיניתו (20606)

נושאי המטלה: יסודות השפה, תנאים,

חומר הלימוד למטרת: ייחדות 1-4

ללאות ופונקציות

משקל המטלה: 3 נקודות

מספר השאלות: 4

מועד אחרון להגשה: 22.11.2025

סמיטר: 2026 א

(ת)

שימוש לב:

- יש להקפיד על שמות המחלקות בדיק כמו שנכתבו.
- יש לטע את התכניות בתיעוד פיניילanganilit בלבך (בתחילה התכנית התיעוד מסביר מה מבצעת התכנית באופן כללי ובמהלך התכניות התיעוד מסביר את הקוד) על פי תקן [8 PEP](#) וכפי שמודגים בדוגמאות ביחידה 1.9 באתר הקורס.
- אין לשמש בחומר מתקדם ובפרט במקרים נטויים מתקדמים ובפרט רשיומות.
- יש לשמש בקבועים היכן שאפשר.
- יש להקפיד על הזחה (איןנטציה - עימוד) נכון, ועל שמות משתנים בעלי משמעות (באנגלית) ולפי המוסכמות בקורס.
- יש להקפיד על פורתט הפלט בדיק כפי שמצוין בשאלת: איות נכון, אותיות גדולות וקטנות, רווחים, וכו'.
- יש טسطר באתר הקורס, כולל קטע קוד שאתה מעתיקם לתוךקובץ המטלה שלכם, אז מרכיבים כדי שהפונקציות שלכם יופעלו ותוכלו לוודא שהכל עובד כמוpected. שימוש לב שלא מגישים עם קוד הטسطר, אלא מוחקים את הקוד הזה אחרי הרצה לבדיקה.
- מטלה שלא תעבור את הטسطר הבסיסי שספקנו לכם, תקבל ציון 0.
- הגשת המטלה נעשית אך ורק בעזרת מערכת המטלות המקוונת שבאתר הקורס.
- אל תשכחו לשמור את מספר האסמכתא שתקבלו מהמערכת לאחר ההגשה.

שאלה 1 (25 נק')

א. כתבו פונקציה בשם `calc` המתקבלת שני מספרים שלמים : `num1`, `num2` ופעולה חשבונית

(חיבור/חיסור/כפל/חילוק) או מטיבוס מחרוזת. הפונקציה תחזיר את תוצאה הפעולה.

במקרה של פעולה חילוק, יש למנוע חילוק ב- 0 ולהחזיר `None`.

במקרה והפעולה זו אינה מייצגת פעולה חוקית, יש להחזיר "invalid".

cohoret הפעולה היא :

```
def calc(num1, num2, op)
```

דוגמאות :

- עבור הפרמטרים (משמאלי לימין) : 4, 10 ופעולה חילוק (//) יוחזר :

~~10 / 4 = 2.5~~

- עבור הפרמטרים (משמאלי לימין) : 3, 5 ופעולה כפל (*') יוחזר :

~~5 * 3 = 15~~

- עבור הפרמטרים (משמאלי לימין) : 0, 12 ופעולה חילוק (//) :

~~None~~

ניתן להניח שהפרמטרים `num2`, `num1`, מייצגים מספרים שלמים (חיובי, שלילי או אפס) והפעולה החשבונית מייצג מחרוזות והוא אחד מהתוויות +, -, *, /. אין צורך לבדוק זאת.

ב. כתבו פונקציה בשם `compute_calcs` המתקבלת מספר שלם חיובי `n`. הפונקציה תקלוט מהמשתמש שלשות כמספר ערכו של `n` : זוג מספרים שלמים ומחרוזת המייצגת פעולה חשבונית (חיבור/חיסור/כפל/חילוק). עבור כל שלשה כזו, יש להדפיס תרגיל המתיצג שלשה זו (כפי שמצווג בדוגמאות לעיל).

במקרה של חילוק ב- 0 יודפס : "Can't divide by zero".

במקרה של פעולה לא חוקית (כלומר פעולה שאין תואמת לאות הפעולה : +, -, *, /),

יש להדפיס "The requested calculation is invalid".

ניתן להניח שהפרמטר `n` מייצג מספר שלם חיובי. אין צורך לבדוק זאת.

cohoret הפעולה היא :

```
def compute_calcs(n)
```

המשך השאלה בעמוד הבא

דוגמיה :

עבור $4=7$, ניתן להציג את הפלט האפשרי הבא :

```
Please insert the first number: 4
Please insert the second number: 5
Please insert an operator: \
The requested calculation is invalid
Please insert the first number: 5
Please insert the second number: 6
Please insert an operator: /
5 / 6 = 0.833333333333334
Please insert the first number: 37
Please insert the second number: 6
Please insert an operator: *
37 * 6 = 222
Please insert the first number: 8
Please insert the second number: 0
Please insert an operator: /
Can't divide by zero
```

שים לב!

- חובה להשתמש בפונקציה שכתבתם בסעיף א.
- הקפידו על רוחחים במקומות הרלוונטיים (לפני ואחרי הפעולה החשבונית וסימן =).
- במקרה של פועלות חילוק אין צורך ליפוט את ההדפסה (למשל, במקרה של תרגיל $3 / 10$, התוצאה היא 3.33333333333335).

שאלה 2 (25 נק')

- א. מספר ארטמיסטרונג (Armstrong Number)¹ הוא מספר שלם בו סכום ספרותיו בחזקת מספר ספרותיו שווה לערך המספר המקורי.

דוגמאות :

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

$$407 = 4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$$

$$8208 = 8^4 + 2^4 + 0^4 + 8^4 = 4096 + 16 + 0 + 4096 = 8208$$

כתבו פונקציה בוליאנית **בשם is_armstrong** המקבלת מספר שלם **num**. הפונקציה תבדוק האם המספר שהתקבל כפרמטר מהו ארטמיסטרונג. אם כן, יחזיר **True**. אחרת, יחזיר **False**.

כוורתה הפונקציה היא :

```
def is_armstrong(num)
```

דוגמאות :

- עבור הפרמטר 153 יחזיר **True**.

- עבור הפרמטר 1634 - יחזיר **False** משום שהמספר שהתקבל אינו חיובי.

- עבור הפרמטר 4321 יחזיר **False** משום שהמספר שהתקבל אינו מהו ארטמיסטרונג.

- עבור הפרמטר 8208 יחזיר **True**.

- ב. כתבו פונקציה אשר מדפיס את כל מספרי ארטמיסטרונג הקיימים ממספרים טבעיים מ- 1 ועד 10000. בנוסף, הפונקציה תחזיר את מספרי ארטמיסטרונג הקיימים בטוויה זה.

כוורתה הפונקציה היא :

```
def count_armstrong_numbers()
```

חויה להשתמש בפונקציה שתכתבם בסעיף א.

לנוחיותכם, ישנים 16 מספרי ארטמיסטרונג בטוויה זה : כל הספרות (1-9),
,371 ,370 ,153 .9474 ,8208 ,1634 ,407

¹ https://en.wikipedia.org/wiki/Narcissistic_number

שאלה 3 (25 נק')

דוחיסת נתונים מאפשרת לנו לעשות שימוש יעיל במדיה (תמונות, קול, וידאו). הדחיסה מאפשרת יעילות גבוהה בפעולות כגון העברת תמונות על גבי האינטרנט ([מקור](#): ויקיפדיה).

נגידר את פרוטוקול הדחיסה הבא:

העבר כלתו המופיע ברכז (שתיים ומעלה), ניתן לדחוס אותו כך שיוופיע פעם אחת ואחריו כמהות המופיעים של התו ברכז. אם התו לא מופיע ברכז, יש לדחוס אותו ללא שינוי.

דוגמאות:

עבור המחרוזת "abcde", המחרוזת הדחוסה היא "a₁b₁c₁d₁e₁" (כל תו מופיע לא בראץ').
עבור המחרוזת "aabccceeeedab", המחרוזת הדחוסה היא "a₂b₁c₃e₅d₁b₁".

כתבו פונקציה בשם `decompressed` המקבלת את המחרוזת הדחוסה `.compress`. הפונקציה תחזיר את המחרוזת המקורי על פי פרוטוקול הדחיסה שתואר לעיל.

הערות:

- ניתן להניה כי המחרוזות **compressed** המתקבלת כפרמטר מיצוג אובייקט מטיפוס **מחרוזת (str)** והתוים במחרוזת מורכבם מאותיות אנגליות קטנות (z-a) וספרות.
 - אין לעשות שימוש באופרטור '**' המוגדר על מחרוזות.
 - ניתן להניה כי שהמספרים המייצגים את כמות התוים הזוהים ברכף שנדחסו מייצגים מספרים חד-ספרתיים (9-2).

שאלה 4 (25 נק')

כתבו פונקציה בשם `count_sub` המקבלת שתי מחרוזות, `s`, `sub`. הפונקציה תחזיר את מספר הפעמים שהמחרוזת `sub` נמצאת במחרוזת `s`. הנירוש שהמחרוזת `sub` אינה ריקה.

שימו לב!

- אין להשתמש באך פונקציה מהמחלקה `str`, למעט מנגנון חיתוך (slicing).
- הפונקציה הסטטיטית `len` מותרת.
- מותר להשתמש באופרטור `+` (שרשור).

דוגמאות :

עבור המחרוזות "ababa", `sub="aba"` יוחזר הערך 2.

עבור המחרוזות "aaaaaa", `sub="aa"` יוחזר הערך 4.

עבור המחרוזת "aba", `sub="abab"` יוחזר הערך 0.

להלן חתימת השיטה :

```
def count_sub(s, sub)
```

הגשה

1. הגשת הממיין נעשית بصورة אלקטרוני בלבד, דרך מערכת שליחת המטלות.
2. יש להגיש קובץ אחד המאגד את כל הפונקציות לשאלות 4-1 בשם `mmn11.py`.
3. ארצו את קובץ הפתרון בקובץ `zip` (ולא `rar`) יחיד ושלחו אותו בלבד.
4. אל תשחו לשמר את מספר האסמכתא **שקיבלתם מהמערכת לאחר ההגשה**. אם לא קיביתם מספר אסמכתא, סימנו שההגשה לא התקבלה.
5. שימו לב, אתם יכולים לשלוח שוב ושוב את המטלה במערכת, אם אתם רוצים לתקן משהו בה. כל הגשה דורשת את ההגשה הקודמת. **אבל עשו זאת אך ורק עד לתאריך ההגשה**. אחרי התאריך, ייחסב לכם כאילו הגשתם באחור, גם אם ההגשה הראשונה הייתה בזמן! כמו כן, אם המנהה הוריד כבר את המטלה שלכם מהמערכת, לא תוכלו לשלוח עותק מעודכן יותר.

בהצלחה