

מבוא למדעי המחשב 67101

תרגיל 3 - לולאות

להגשה בתאריך 11/11/2015 בשעה 22:00

בתרגיל זה נתרגל שימוש בלולאות ומשתנים. התרגיל מורכב ממספר משימות בלתי תלויות.

דגשים

- כתבו את כל הקוד שלכם בתוך הקובץ `ex3.py`.
- החתימה של הפונקציות צריכות להיות זהה במדויק לחתימה שמתוארת במשימות.
- לנוחותכם, מסופק קובץ שלד הכולל את כל חתימות הפונקציות שיש לממש. ניתן להוסיף לקובץ פונקציות נוספות במידת הצורך.
- סגנון:
 - הקפידו על תיעוד נאות.
 - בחרו שמות משתנים משמעותיים.

חלק א' - קבלת קלטים מהמשתמש

1. קבלת רשימה כקלט

ישנן דרכים שונות לקבלת קלט. דרך אחת היא להשתמש בפונקציה `input()` אשר מקבלת שורת קלט ממירה אותה למחרוזת. לדוגמא, המשתמש מזין ערך `'hello'` למשתנה `str_element` ולאחר מכן ניתן להשתמש במשתנה זה:

```
str_element=input()
print(str_element)
```

עבור הקלט `hello`, התכנית הנ"ל תדפיס

```
hello
```

a. כתבו פונקציה שמקבלת קלט מהמשתמש בלולאה. הלולאה עוצרת כאשר המשתמש מכניס מחרוזת ריקה. הפונקציה מחזירה רשימה של כל הערכים שהמשתמש הכניס כקלט. חתימת הפונקציה:

```
def create_list():
```

ניתן להשתמש בפונקציה `.append()`.

לדוגמא עבור הקלט:

```
as
sig
nm
ent
3
```

הפונקציה תחזיר את הרשימה

```
['as', 'sig', 'nm', 'ent', '3']
```

הנחות על הקלט:

- אם המשתמש לא מכניס קלט, הפונקציה מחזירה רשימה ריקה.

b. כתבו פונקציה שמחזירה את השירשור (concatenation) של איברי רשימה כלשהיא הניתנת כקלט.

חתימת הפונקציה:

```
def concat_list(lst_str):  
    כאשר lst_str היא רשימה של מחרוזות.  
    לדוגמא עבור הקלט  
    ['as', 'sig', 'nm', 'ent', '3']  
    הפונקציה צריכה להחזיר  
    'assignment3'  
    ניתן לבצע שירשור שתי מחרוזות ע"י האופרטור +. לדוגמא,  
    'or'+'ange'  
    יחזיר  
    'orange'  
    במשימה זאת לא ניתן להשתמש בפונקציה .join().
```

2. קבלת רשימת מספרים כקלט

כתבו פונקציה שמקבלת רשימה של מספרים ממשיים float ומחזירה את הממוצע שלהם.
חתימת הפונקציה:

```
def avr(num_list):
```

לדוגמא עבור הקלט:

2.7, 5.3, 2.5, 0, 6.5

הפלט יהיה:

3.4

מכיוון ש $3.4 = (2.7 + 5.3 + 2.5 + 0 + 6.5) / 5$
הערה: לא ניתן להשתמש בפונקציה sum.
הנחות קלט:

- הקלט מהמשתמש הוא מספרים ממשיים בלבד.
- במידה ולא מכניסים אף קלט, הפונקציה מחזירה None.

חלק ב' -לולאות על רשימות

1. הסטה מעגלית

הסטה מעגלית של רשימה ב-d מקומות, היא הזזה קדימה של כל האיברים d איברים קדימה כך שאיברים שחורגים מסוף הרשימה מושמים חזרה בתחילתה. כלומר, איבר עובר מאינדקס i לאינדקס $(i+d)\%k$ כאשר k הוא אורך הרשימה.

לדוגמא הסטה של $d=1$ עבור הרשימה:

```
['a', 'b', 'c', 'd']
```

תחזיר

```
['b', 'c', 'd', 'a']
```

כיוון שהאינדקסים ברשימה המקורית ממופים לאינדקסים ברשימה החדשה באופן הבא:

0 <- 1

1 <- 2

2 <- 3

3 <- 0

פרמוטציה ציקלית של רשימה היא הסטה ציקלית של הרשימה בגודל d כלשהו.

לדוגמא:

```
['a', 'b', 'c']
```

יש 3 פרמוטציות ציקליות:

```
['a', 'b', 'c']
```

```
['c', 'a', 'b']
```

```
['b', 'c', 'a']
```

כתוב פונקציה שמקבלת שתי רשימות ומחזירה true אם רשימה אחת היא פרמוטציה ציקלית של השנייה, אחרת הפונקציה מחזירה false.

חתימת הפונקציה:

```
def cyclic(lst1, lst2):
```

כאשר lst1 ו-lst2 הן רשימות.

אם שתי הרשימות אינן באורך זהה אז הפונקציה מחזירה False.

2. היסטוגרמה

נתונה רשימה של מספרים שמכילה רק מספרים שלמים בטווח 0...n-1. היסטוגרמה של רשימה בטווח של מספרים היא מספור של הפעמים שכל איבר בטווח המספרים הופיע ברשימה. כתוב פונקציה שמקבלת רשימה של מספרים שלמים ומספר שלם n, ומחזירה את ההיסטוגרמה שלה.

חתימת הפונקציה:

```
def hist(n, list_num):
```

לדוגמא, נניח שהטווח הוא 0...4 עבור ההיסטוגרמה:

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

הפלט יהיה

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

מכיוון ש:

0 הופיע 0 פעמים

- 1 הופיע פעם אחת
- 2 הופיע 2 פעמים
- 3 הופיע 0 פעמים
- 4 מופיע 3 פעמים

הערה: לא ניתן להשתמש במודול numpy

שימו לב:

- אורך הרשימה המוחזרת הוא n.
- מספרים שלא מופיעים ברשימה, הערך בהיסטוגמה (לפי האינדקס המתאים) הוא 0.

הנחת קלט: list_num מורכבת מורכבת ממספרים שלמים בטווח (n-1,0) בלבד.

חלק ג' - לולאות מקוננות

1. פירוק לגורמים

כתוב פונקציה שמקבלת מספר שלם חיובי n ומחזירה את רשימה של כל הגורמים הראשוניים שלו (מהקטן לגדול).

חתימת הפונקציה:

```
def fact(n):
```

לדוגמא:

```
fact(24)
```

הפלט יהיה

```
[2, 2, 2, 3]
```

מכיוון ש $24=3*2*2*2$.

הערה: המשפט היסודי של האריתמטיקה אומר שישנו פירוק יחיד למספרים ראשוניים. כלומר במקרה הנ"ל 2 ו-3 הם הגורמים הראשוניים היחידים של 24.

ניתן לבצע זאת ע"י מעבר סדרתי של המספרים (מחלקים) עוקבים מ-2 עד n, ועבור כל מספר לבדוק כמה פעמים ניתן לחלק את n במספר עצמו ללא שארית.

שאלה למחשבה (ללא ניקוד): האם ניתן להקטין את כמות המספרים הנבדקים? איך ניתן לייעל את מציאת הגורמים?

הנחת קלט: ניתן להניח שהקלט הוא מספר חיובי שלם.

2. מכפלה קרטזית

מכפלה קרטזית של שתי קבוצות היא אוסף כל הזוגות בהם איבר אחד הוא בקבוצה הראשונה והאיבר השני הוא בקבוצה השנייה. כתוב פונקציה אשר שמקבלת שתי רשימות של מחרוזות ומחזירה את המכפלה הקרטזית שלהם.

חתימת הפונקציה:

```
def cart(lst1, lst2):
```

כאשר lst1 ו-lst2 הן רשימות של מחרוזות.

לדוגמא עבור

```
['a', 'b', 'c']  
['m', 'n']
```

הפלט יהיה:

```
[['a', 'm'], ['a', 'n'], ['b', 'm'], ['b', 'n'], ['c', 'm'], ['c', 'n']]
```

כלומר הפלט הוא רשימה מקוננת של כל זוגות של איברים (רשימה של שני איברים) כך שהאיבר הראשון הוא ב-lst1, והאיבר השני הוא ב-lst2.

שימו לב שאורך שתי הרשימות לא חייב להיות זהה.

סדר האיברים ברשימת הפלט אינו חשוב.

הערה: לא ניתן להשתמש במודול numpy

הנחה על הקלט: אם לפחות אחת הרשימות ריקה, אז הפונקציה תחזיר רשימה ריקה.

3. זוגות שווי סכום

כתוב פונקציה שמקבלת רשימה של מספרים שלמים, ומספר שלם נוסף n, ומחזירה כל הזוגות (רשימות באורך 2) השונים ברשימה שסכומם שווה בדיוק ל-n.

חתימת הפונקציה:

```
def pair(n, num_list):
```

כאשר n הוא מספר שלם ו-lst1 היא רשימה של מספרים.

לדוגמא:

```
n=5  
num_list=[-2, 3, 7, 11, 2, 6]
```

הפונקציה תחזיר שני זוגות [3,2] ו [7,-2] מכיוון שסכומם שווה ל-5. במידה ואין אף זוג כזה הפונקציה תחזיר None.

הנחות קלט: ניתן להניח שכל שהרשימה מכילה מספרים בלבד ושהם שונים זה מזה (כלומר כל זוג מופיע פעם אחת בלבד).

הגשת התרגיל

בשלב זה אמורים להיות לכם הקבצים:

1. `ex3.py`

2. `README` (כפי שמופרט בנהלי הקורס)

עליכם ליצור קובץ `zip` בשם `ex3.zip` שיכיל את שני הקבצים הנ"ל.

בהצלחה!