

## תרגיל בית 3

1. (20%) לכל מספר טבעי  $n$ , פונקציית המספרים הראשוניים  $\pi(n)$  מציינת את מספר המספרים הראשוניים הקטנים או שווים ל  $n$ . למשל,  $\pi(2)=1$ ,  $\pi(3)=2$ ,  $\pi(10)=4$ ,  $\pi(20)=8$ ,  $\pi(100)=25$ . ראו [כאן](#) מחשבון שמחשב את  $\pi(n)$ . [משפט המספרים הראשוניים](#) קובע ש  $n/\ln(n)$  הוא קירוב של  $\pi(n)$  והקירוב הזה משתפר ככל ש  $n$  גדול יותר.

יהי  $\max=10^9$ . כיתבו תוכנית primes\_density.py המייצרת 100000 מספרים אקראיים בין  $\max/2$  ל  $\max$ , מחשבת ומדפיסה את אחוז המספרים הראשוניים מתוך מספרים אלה. השוו את התוצאה עם  $1/\ln(10^9)=0.0483$ . יש להשתמש בפונקציה is\_prime שבמצגת 5.

דוגמה לפלט:

```
density of primes: 0.0481
expected density: 0.0483
```

2. (40%) בשאלה זאת נחשב מדדים סטטיסטיים של נתונים מהקלט. את התוכנית יש לכתוב בתוך קובץ stats.py.

א. כתבו פונקציה isfloat שמקבלת מחרוזת ומחזירה True אם המחרוזת מייצגת מספר עשרוני ממשי, ו-False אם לא. מחרוזת היא מספר ממשי אם...  
(i) היא מכילה רק תווים מהמחרוזת "0123456789-+."  
(ii) היא מתחילה בסיפורה או ב + או ב - (מינוס) ומסתיימת בסיפורה  
(iii) היא מכילה לכל היותר נקודה עשרונית אחת  
(iv) הסימנים + ו- יכולים להופיע רק בתו הראשון  
(v) אם מופיע בהתחלה + או - אז התו הבא חייב להיות סיפורה  
(vi) אם הסיפורה הראשונה במחרוזת היא 0 וזאת לא הסיפורה האחרונה, אז התו הבא חייב להיות . (נקודה עשרונית) (למשל, 0.123 היא מחרוזת חוקית, אבל 0123 לא.)

דוגמאות למחרוזות חוקיות:

0.123, -0.123, +0.123, -4.01, 12345, +3.05, 3.00, 0, -0, +0

דוגמאות למחרוזות לא חוקיות:

.123, +.5, 54.6.7, +123+5, 45., 00.123, 001234, 123a4

ב. כתבו פונקציה string\_to\_list שמקבלת מחרוזת המורכבת ממספרים ממשיים המופרדים ברווחים ומחזירה רשימה המכילה את המספרים האלה. אם אחד המספרים לא חוקי (ר' סעיף א') הפונקציה תחזיר None.

ג. כתבו פונקציה mean המקבלת רשימת מספרים ומחזירה את הממוצע שלהם.  
ד. כתבו פונקציה sd המקבלת רשימה של מספרים ומחזירה את [סטיית התקן](#) שלהם.  
ה. כתבו פונקציה median המקבלת רשימה של מספרים ומחזירה את [החציון](#) של המספרים ברשימה. לדוגמה, החציון של [2,5,7,8,9] הוא 7, החציון של [2,5,7,8] הוא 6 (הממוצע של 5 ו-7). שימו לב שכדי לחשב את החציון יש למיין תחילה את הרשימה.

**יש לכתוב את הפונקציה בעצמכם ולא להשתמש בפונקציית ספרייה.**

ו. כתבו תוכנית stats.py שקוראת מקובץ בשם numbers.txt מספרים שלמים המופרדים ברווחים ומדפיסה לקובץ בשם stats.txt את הממוצע שלהם, סטיית התקן שלהם ואת החציון שלהם.

מצורפים קובצי קלט ופלט לדוגמה

התוכנית תעצור אם הקלט אינו תקין, כלומר, אם הקלט אינו מורכב ממספרים. במקרה כזה התוכנית תדפיס הודעה illegal input.

3. (20%) בקובץ matrix.py כיתבו תוכנית לעבודה עם מטריצות לפי ההנחיות הבאות:  
"מטריצה" היא רשימה של רשימות מספרים שכולן באותו גודל.

למשל,  $[[1,2,3],[4,5,6],[7,8,9]]$  היא מטריצה, אבל  $[[1,2,3],[4,5],[6,7,8]]$  אינה מטריצה

א. השתמשו ב list comprehension כדי לכתוב פונקציה matrix\_scalar\_mult שמקבלת מטריצה A ומספר c ומחזירה את תוצאת הכפל של A ב c (כלומר, את cA). אין להשתמש בלולאות! (cA היא המטריצה המתקבלת מכפל כל אחד מאברי A ב c)

ב. השתמשו ב list comprehension כדי לכתוב פונקציה matrix\_add שמקבלת שתי מטריצות A ו B ומחזירה את  $A+B$  (כלומר, על הפונקציה להכיל פקודה אחת בלבד. אין להשתמש בלולאות! (אפשר להניח של A ו B אותו מספר שורות ואותו מספר עמודות)

ג. השתמשו ב list comprehension כדי לכתוב פונקציה matrix\_mult שמקבלת שתי מטריצות A ו B ומחזירה את  $AB$  (כלומר, על הפונקציה להכיל פקודה אחת בלבד. אין להשתמש בלולאות! (אפשר להניח שהפרמטרים של הפונקציה חוקיים, כלומר, מספר העמודות ב A זהה למספר השורות ב B).

ד. כיתבו פונקציה identity\_matrix שמקבלת מספר טבעי חיובי n ומחזירה את מטריצת היחידה  $I_n$  מסדר n. (מטריצת היחידה  $I_n$  היא המטריצה חח שכל אברי האלכסון הראשי שלה הם 1 ושאר האברים הם 0). יש ליצור את המטריצה  $I_n$  באמצעות list comprehension ולא לולאה.

ה. נייצג פולינום ע"י רשימה של מספרים ממשיים כך שהאיבר במקום ה i מייצג את המקדם של החזקה ה i. למשל, הרשימה  $[2,1,0,-3,2.5]$  מייצגת את הפולינום  $2x^4 - 3x^3 + x^2 - 3x + 2.5$ . כיתבו פונקציה matrix\_polynom שמקבלת רשימה p, המייצגת פולינום, ומטריצה ריבועית A ומחשבת את תוצאת ההצבה של A בפולינום p. למשל, עבור הפולינום שבדוגמה, הפונקציה תחשב את  $2I_n + A - 3A^3 + 2.5A^4$ , כאשר n הוא הסדר של A (כלומר, A היא חח).

ו. כיתבו פונקציה print\_matrix שמקבלת מטריצה וקישור לקובץ פלט ומדפיסה אותה כך שכל שורה במטריצה מופיעה בשורה נפרדת בקובץ הפלט.

ז. כיתבו תוכנית שקוראת מקובץ קלט matrix\_input.txt מטריצה ריבועית A ורשימת מספרים המייצגת פולינום ומדפיסה לקובץ matrix\_output.txt את תוצאת ההצבה של A בפולינום. כל שורה במטריצה נמצאת בשורה נפרדת בקובץ וגם הפולינום נמצא בשורה נפרדת, מיד אחרי המטריצה. אפשר להניח שהמטריצה ריבועית, אבל גודל המטריצה לא ידוע מראש.

דוגמה לקלט:

```
1 0 -1
0 2 1
3 1 0
2 1 0 -3 2.5
```

3 השורות הראשונות מייצגות את המטריצה  $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 3 & 1 & 0 \end{bmatrix}$  והשורה הרביעית את

הפולינום  $2I + A - 3A^3 + 2.5A^4$ .

אפשר להניח שהקלט תקין. יש להדפיס את הפלט עם שתי ספרות אחרי הנקודה העשרונית

מצורפים 6 זוגות קבצים של דוגמאות קלט ופלט.

הנחיות הגשה:

- 1 יש להגיש תוכניות שרצות ללא שגיאות. תוכנית שתגוש עם שגיאות תקבל לכל היותר חצי מהנקודות.
- 2 יש לכתוב הערות לתוכנית: `docstring` בתחילת כל פונקציה, הסבר קצר בתחילת התוכנית, הסבר בתחילת לולאות.
- 3 אין להשתמש במודולים מלבד מודולים סטנדרטיים כמו `math`, `random`, `sys`, `timeit`
- 4 יש לפתור כל שאלה בקובץ נפרד עם סיומת `.py`.
- 5 יש להגיש את כל הקבצים בקובץ אחד מכוון עם סיומת `.zip`. שם קובץ ה `zip` צריך להיות מספר הת"ז שלכם ומספר עבודת הבית. למשל, `22222222_hw3.zip`
- 6 כל קובץ יתחיל בהערה ובה המידע הבא:
  - א. שם הסטודנט
  - ב. מס' תעודת זהות
  - ג. מספר דף התרגילים
  - ד. שם התוכניתלמשל, עבור תרגיל 3 בדף 3:

```
"""
Student: Julius Caesar
ID: 111112113
Assignment no. 3
Program: matrix.py
"""
```

שימו לב: יש להקפיד על הנחיות ההגשה האלה. הגשה שלא בדיוק בפורמט הזה לא תקבל את מלוא הנקודות ואף עלולה להיפסל.