

## : תרגיל מסכם

### חלק א: מחשבון ממש מושכלל

מחשבון זה הוא מאוד מושכלל ומכל קצת יותר מחשבון כיס רגיל. תוכנית המחשבון תהיה דומה מאוד כל שעליה לעשות הוא קלוט נתונים מהתואנה – Interpreter Python של –Interpreter Python של Python-ל Python-ל.

הפעולות שיכללו במחשבון הן:

**חיבור חיסור כפל חילוק חזקה מודולו מקסימום מינימום ממוצע שלילה עצרת סימן ! ~ @**

												עוצמת קידומו ת
				1 1 2 2								
				ממוצע ממוצע ממוצע								מיקומית אופרטור ר
X!	~X	X@X	X&X	X\$X	X%X	X^X	X/X	X*X	X-X	X+X	נראות שימוש אופרטור ר	

בטבלה זו, העוצמה קובעת אילו פעולות חזקות יותר מאחרות – למשל, כפל חזק יותר לחבר (כפי כפל בא לפני חיבור), ושלילה בא להפני כולם, לפי עוצמת הקידומת שרשומה בטבלה. פעולות הממוצע, מינימום ומקסימום מקבלות שני אופרנדים, וביצוען עליהם את הממוצע, מקסימום ומינימום בהתאם. למשל:

- נדרש להחזיר 3, כי זה הממוצע בין 2 ל 4 @ 2.
- נדרש להחזיר 7, כי 7 גבוהה מ 2 \$ 2.

כאשר מופיעות מספר פעולות אחת אחרי השניה, ולכל הפעולות עוצמה זהה, יש לבצע את הפעולות בסדר בו הן מופיעות. כמו כן, המחשבון צריך לכלול טיפול בסוגרים (רק סוגרים מסווג אחד), כמו למשל [ ], {}, () . רק () . כאשר סוגרים קודמות לכל

על הקוד לקבל כל קלט מבלי לקרים, ולהשיב בהודעת שהגיה אינפורטטיבית במידה והקלט לא תקין או התשובה אינה תקינה לחישוב (לדוגמא חילוק ב 0).

הקוד צריך לדעת לטפל בכל המספרים המשמשים.

### דוגמאות והסבירים

הקידימות של חיבור היא 1, הקידימות של כפל היא 2.

לכן במקרים דומים  $4+5*2 = >$  קודם כל תבצע פעלת הכפל  $2*5$  ורק לאחר מכן פעלת החיבור.

הקידימות של מקסימום הוא 5, הקידימות של עצרת היא 6. לכן במקרה הבא:  $7@3! <$  קודם כל תבצע פעלת העצרת  $!3$  ורק לאחר מכן פעלת המקסימום.

כאשר יש יותר פעולה אחת בעלת אותה קידימות על אותו המספר, נחשב את הפערון משמאל לימין. לכן במקרה הבא:  $-3\sim$  קודם כל תבצע פעולה השלילה  $-3$  ורק לאחר מכן פעלת העצרת.

עליכם להבדיל בין סימון מספר שלילי לבין האופרטור של חיבור

חסוך : 3-3

סימן: 3-+3

במידה ויש מספר בעל יותר מסימן מספר שלילי אחד ברכף הוא ישנה בהתאם היפוך ל- 3 3

!!!ב berhasilה

## **או מגה 2**

### שלילה (טילדה (~) ~ ') והשימוש בה

שלילה תוכל להיצמד אך ורק ליד מספר

	תיקין לא תיקין
4~@3	~3
3~	~-3
~~---3	~~~~~3
~~3	3+~-3

### שימוש האופרטורים

למעט בחזקה  $\mathbf{math}$  אסור להשתמש באך ספריה חיצונית של פיתון בין היתר. את כל פועלות האופרציה אתם נדרשים לבצע בעצמכם  
**על מנת לבצע את הפעולה המתמטית `sosk`) בלבד חזקה. בחזקת תוכנו להשתמש בפעולת השמורה \*\*\*.**

### הנחיות:

- `PEP8` הקפיד על כתיבה נכונה, ולפי חוקי הפורמט של 8.
- הקפד על חוב טכנולוגי נמוך, הקוד-Amoor להיות כתוב בצורה שיעמוד עם שינויים בעתיד לדוגמא . האם הקוד יימוד בהוספה של עד אופרטור ווכו.
- יש שיבת על מקרה קצה היא חשובה, התוכנית-Amora להיות מוכנה לכל ביתו .
- יש לחלק את התוכנית למודולים לפי הצורך, ולתעד הכל בצורה תקינה וברורה .
- קוד קרייא וחכם הוא קוד בריא .
- תשתפו לשאר חברי היכתה במקורי קצה שהקשו על הקוד שלכם על מנת לעזור לשפר את הקוד . • עלי .
- התוכנית לקלט כל קלט בלי לקרויס בשום מצב, ולהשיב בהודעת שגיאה אינפורטטיבית במידה והקלט לא תקין .
- כדי שלמדתם בשיעור, `git` יש לעבוד בצורה נכונה עם .

### חלק ב': בדיקה עצמית של התרגילים

- pytest לאחר שהתרגיל הושלם לפי ההוראות הגיע הזמן לבדוק אותו באמצעות `pytest`.  
ולהתיקן אותו, `pip` מ `pytest` תחילת עליך להוריד את החבילת `pytest` ממצא בגוגל הוראות מפורטות לעובדה Um .  
: באמצעות חבילת זו ממש לפחות את הטסטים הבאים ותבדוק את הקוד שלא באמצעות " עד 5 שגיאות סינטקס פשוטות, לדוגמה: " `3^*2` .  
• מחירות ג'יבריש .  
• מחירות ריקה .  
`TABS` יש לשים לב גם ל `white space` מחירות .  
עד 15 משויות פשוטות, אשר ידקנו את כל האופרטורים לפחות פעם אחת, לדוגמה: `3^2` שים לב .  
שלא מספיק שהקוד לא יקרוס, אלא הוא צריך להחזיר תשובה נכונה .  
עד 20 משויות מורכבות תקינות הכלולות סוגרים, רווחים, ומספר אופרטורים וסדר פעולות נכון. על .  
כל משווה להיות באורך 20 תוים לפחות. שוב, לא מספיק שהקוד לא יקרוס, אלא הוא חייב להחזיר תשובה נכונה .

שים לב שהוא תרגיל מסכם ונדרשת רמת ביצוע גבוהה, הקוד-Amoor לעבוד בכל קלט ובפרט עבור את כל המבחןים שתכתב ועומד בכל הדרישות. הקוד-Yibok על תוכנו, צורת כתיבתו, ועל

עובדתו.

!!!בהצלחה

**או מגה 2**

**: גרסאות**

- Python-3.7
- Pytest-6.2

**: הגשה**

classroom  
עד התאריך 22.12 בשעה 21:00  
בלי איתורים !!!

**!!!בהצלחה**