המכללה האקדמית להנדסה סמי שמעון

עבודת הגשה 1

- יש להגיש את העבודה עד **2019-12-08** דרך Moodle בלבד, בקובץ python בודד (סיומת py).
- הגשה בבודדים בלבד! (אין הגשה בזוגות, שלשות וכדומה).
- אין להעתיק: העתקה תגרום לפסילת העבודה של המקור והמעתיקים והעברה לוועדת משמעת.
- יש לוודא התאמה שם שמות הפונקציה, בדגש על אותיות גדולות וקטנות.
- קראו וודאו עבור כל פונקציה את צורת הקלט (פרמטרים) וצורת הפלט (החזרה או הדפסה).
- עבור כל פונקציה, קיימות מספר דוגמאות הרצה. ודאו שהתוצאה מהפונקציה שלכם מתאימה לדוגמאות האלה.
- לכל פונקציה יש לכתוב docstring מתאים, ולכתוב לכל פרמטר מה הוא מקבל ומה ההנחות עליו.
- קובץ שלא מסוגל לרוץ עקב טעות syntax גורר אפס אוטומטי עבור השאלה.
- לשאלות, תפנו למייל כשבפנייה חובה לציין שם פרטי ושם משפחה והסבר מפורט של השאלה.
svetazam+sce@gmail.com
- קראו טוב את השאלות וודאו שאתם מבצעים היטב את השאלה.

שאלה 1

כתבו פונקציה בשם $parseStr$ המקבלת מחרוזת המייצגת מספר דצימלי שלם ומחזירה את אותו המספר אבל כמספר.

אין להשתמש בהמרה ישירה של Python אלה יש לכתוב אותה.

רמז: השתמשו בפונקציה ord, ואל תשכחו לטפל במספרים שליליים.

```
>>> parseStr('1000')
1000
>>> parseStr('15')
15
>>> parseStr('0')
0
>>> parseStr('-10')
-10
```

שאלה 2

כתבו פונקציה בשם $smallHash$ המקבלת מערך לא ריק של מספרים שלמים בטווח $[0,255]$ ומחזירה תוצאת חישוב hash על המערך לפי הפונקציה הבאה:

$$f(a_0) = a_0 \quad ; \quad f(a_n) = \left(\left(31 \cdot f(a_{n-1})\right) xor\ a_n\right)\ and\ 0xFF$$

אין להשתמש ברקורסיה, אלה רק בלולאה.

רמז: כדי לבצע את הפעולות xor, and תסתכלו כאן.

```
>>> smallHash([5])
5
>>> smallHash([5,6])
157
>>> smallHash([5,6,19])
16
>>> smallHash([7,6,19])
18
```

שאלה 3

כתבו פונקציה בשם $interceptPoint$ המקבלת 4 מספרים ממשיים בשם $m1, n1, m2, n2$ המייצגים את הפרמטרים של שתי משוואות ישרים $y = m1 \cdot x + n1$. עליכם להחזיר $None$ אם אין נקודת חיתוך (ישרים מקבילים או מתלכדים), ולהחזיר tuple של נקודת החיתוך בצורה (x, y) אם הישרים חותכים זה את זה.

רמז: תנסו לפתור כמערכת 2 משוואות מעל 2 משתנים מעל הפרמטרים הללו.

```
>>> interceptPoint(5,4,5,9)
None
>>> interceptPoint(2,4,5,-2)
(2.0, 8.0)
>>> interceptPoint(3,11,5,1)
(5.0, 26.0)
>>> interceptPoint(5,4,5,4)
None
```

שאלה 4

כתבו פונקציה בשם $factorSum$ המקבלת מספר שלם גדול מ-1, ומחזירה את סכום כל הגורמים הראשוניים שלו. זכרו שהמספר 1 אינו ראשוני.

לדוגמה: $f(60) = f(2^2 \cdot 3 \cdot 5) = 2 + 3 + 5 = 10$.

```
>>> factorSum(2520)
17
>>> factorSum(625)
5
>>> factorSum(81)
3
>>> factorSum(221)
30
```

שאלה 5

כתוב פונקציה **רקורסיבית** בשם $bracketsCheck$ המקבלת מחרוזת, ומחזירה אמת אם סדר הסוגריים **העגולות** הוא נכון ומלא בכל המחרוזת, אחרת מחזירה שקר. שימו לב, שיש לבדוק שבכל מקום במחרוזת מספר הסוגריים השמאליים גדול שווה מהסוגריים הימניים, ובסוף המחרוזת מספר הסוגרים שווה.

רמז: מותר להשתמש בפונקציית עזר, אבל היא חייבת להיות פנימית (nested).

```
>>> bracketsCheck('()')
True
>>> bracketsCheck('())')
False
>>> bracketsCheck('())(')
False
>>> bracketsCheck('(a))(')
False
>>> bracketsCheck('a(bc{d)(8)')
True
>>> bracketsCheck('a(bc[]][)')
True
```

שאלה 6

כתוב פונקציה **רקורסיבית** בשם $printBin$ המקבלת מספר שלם חיובי, ומדפיסה את המספר בבינארי. אין להשתמש בפונקציה bin או ב-$str.format$ או בפונקציה מובנת אחרת – יש לממש רקורסיבית את הפונקציה.

אנו לא מעוניינים שבהדפסה כל סיבית תהיה בשורה חדשה, ולכן מצאו מה יש להעביר ל-`print` כדי שלא יוסיף שורה חדשה.

רמז: כדי להמיר מבסיס דצימלי לבסיס אחר, צריך לחלק ולראות שארית ולסדר בצורה נכונה.

```
>>> printBin(5)
101
>>> printBin(50)
110010
>>> printBin(800)
1100100000
>>> printBin(895623)
11011010101010000111
```

שאלה 7

כתוב פונקציה **רקורסיבית** בשם $evenFactorial$ המקבלת מספר שלם חיובי n, ומחזירה את מכפלת כל המספרים **הזוגיים** בטווח $[1, n]$.

```
>>> evenFactorial(5)
8
>>> evenFactorial(6)
48
>>> evenFactorial(8)
384
>>> evenFactorial(3)
2
>>> evenFactorial(7)
48
```