

תרגיל 2: פונקציות

תאריך פרסום: 7.11.2024

תאריך הגשה: 3.2.2025

מתרגלת האחראית: עומרית אבטל

משקל תרגיל: 2 נקודות

מטרות העבודה: שימוש בפונקציות ועובדיה עם מחרוזות

dagshim לתרגיל 2

- בשימוש תרגיל זה עליכם להשתמש אך ורק בפקודות הבאות, כלומר אין להשתמש בפקודות אחרות, אפילו אם הן נלמדו בשיעור.

- תנאים: in, if, elif, else, in, range, break, continue, for, while,新格局。
- לולאות: for, while, continue, break, in, range.
- ניתן להשתמש במשתנים מספריים, בוליאניים, תווים, מחרוזות ורשימות.
- אין להשתמש בסוגי משתנים אחרים (כגון מילונים, tuple, סטים וכו').
- הfonקציות האריתמטיות (+, -, *, **) והלוגיות (or, and וכו') של פיתון.
- מתוך הfonקציות המובנות של פיתון ניתן להשתמש רק באלו: .len, chr, str, ord, int, bool, float, list, pow, type, range, min, max, print, input,sum
- כל השיטות של String (אלא אם כן נאמר אחרת).
- השיטות של List (.append, .insert, .index, .pop) של List.
- ניתן להוסיף פונקציות עצם.
- חובה לכתוב docstring (כפי שלמדתם בתרגול) לכל פונקציה.

שאלה 1:

נגיד ערימה של פנק'יקים כערימה בה הפנק'יקים מסודרים אחד מעל השני בסדר אקראי.
נגיד ערימת פנק'יקים מושלמת כערימה בה הפנק'יקים מסודרים בסדר יורד כך שהקטן ביותר נמצא
למעלה והגדול ביותר נמצא למטה.

נגיד ערימה של פנק'יקים כרשימה של מספרים טבעיות, המספרים ברשימה מייצגים את גודלים של
הפנק'יקים. במקומות הראשונים של הרשימה (אינדקס 0) יש את הפנק'יק העליון בערימה, ובמקומות
האחרון יש את הפנק'יק התחתון בערימה.

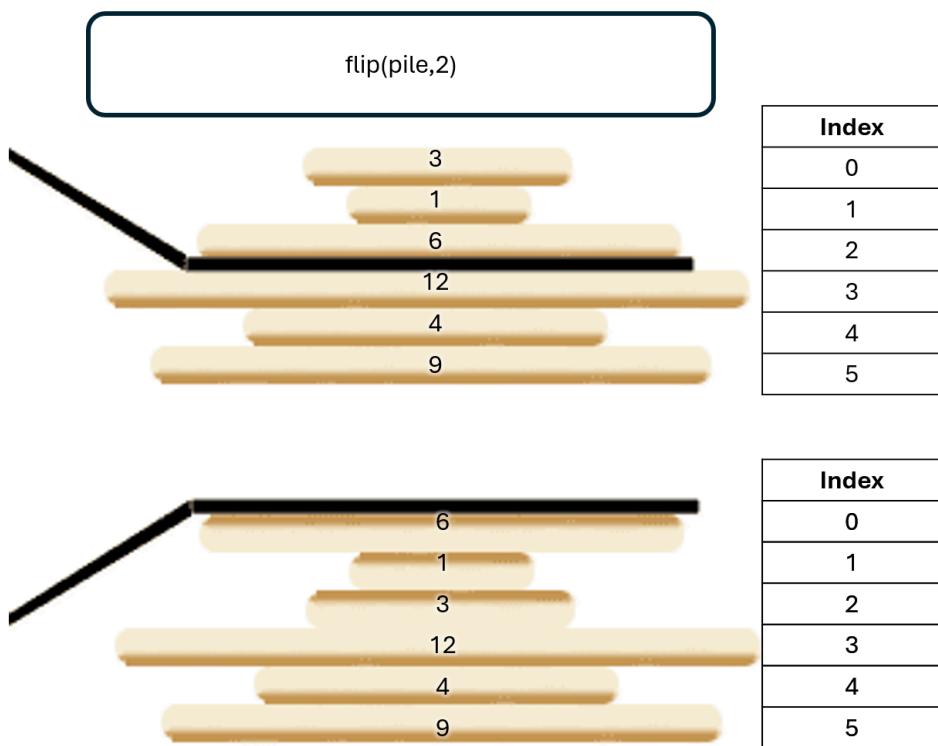
הדרך היחידה לשנות את מקום הפנק'יקים בערימה הוא להכניס את המריה בין שני פנק'יקים ולהפוך
את כל מה שנמצא מעלה.

 סעיף א':

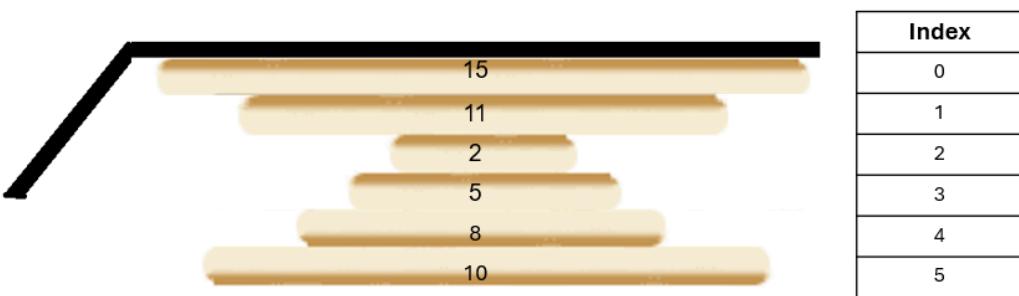
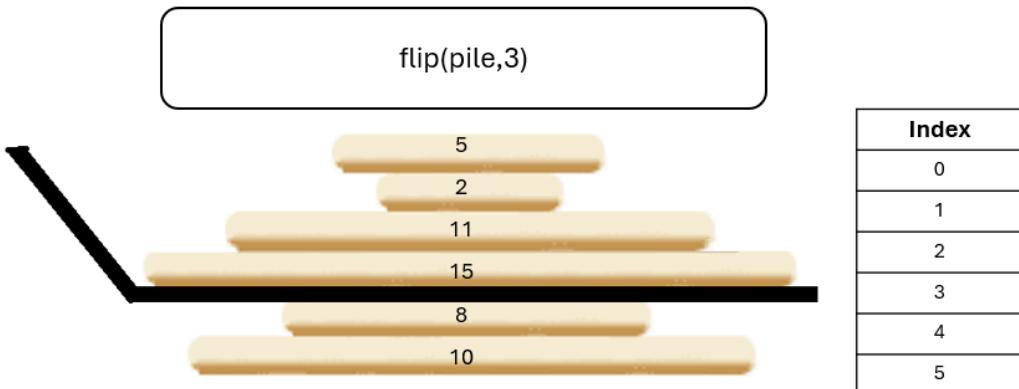
משמעות פונקציה בשם (pile,index) flip

הfonקציה מקבלת כקלט רשימה המציגת ערימת פנק'יקים המוצגים כמספרים טבעיות ואינדקס
(index) שמתחילה הערימה (אינדקס 0) ועד אליו יש להפוך את הערימה.
הfonקציה מחזירה רשימה בה בוצע היפוך של תת-הערימה מתחלת הערימה ועד האינדקס שצין
(כולל).

לפניכם דוגמה של ביצוע פעולה flip :



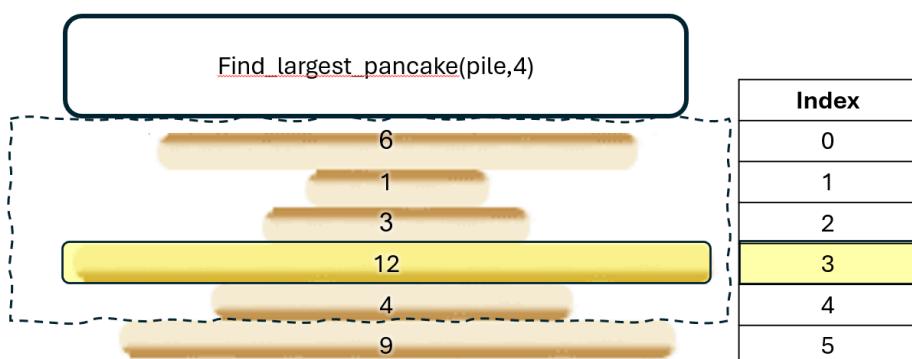
דוגמה נוספת:

עיף ב':

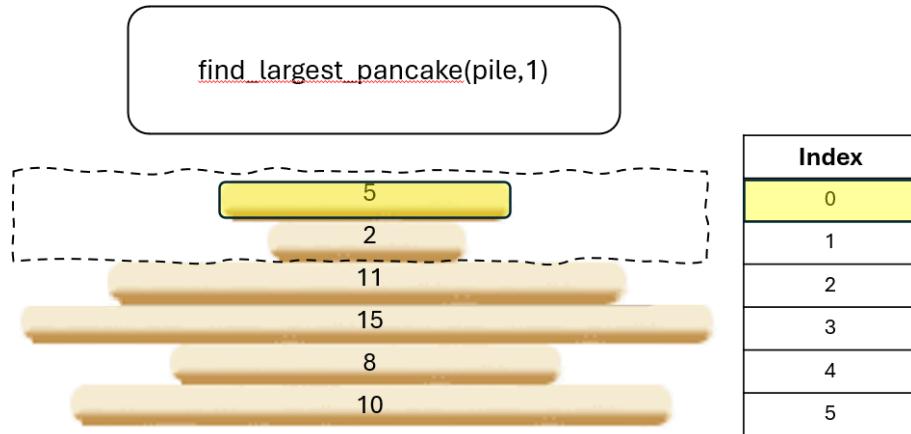
מימוש פונקציה בשם `find_largest_pancake(pile, index)`

הfonקציה מקבלת כקלט רשימה (`pile`) המייצגת עPILEת פנקיקים ומספר טבעי המיצג אינדקס ברשימה. הפונקציה מחזירה מספר טבעי המיצג את האינדקס של הפנקיק הכי גדול בתת רשימה המורכבת מהפנקיקים מהתחלת הערימה ועד האינדקס המציין (`index`) (כולל אותו).

לפניכם דוגמה של ביצוע פעולה `find_largest_pancake`, בצהוב מסומנים הפנקיק הכי גדול בתת הרשימה המוקפת והאינדקס שלו:



הפעולה במקורה זהה תעבור על תת הרשימה המוקפת ותחזיר את הערך 3.



ובמקרה זהה תעבוד הפונקציה על תת הרשימה המוקפת ותחזיר את הערך 0.

סעיף ג':

ממשו את הפונקציה `pancake_sort(pile)`

הfonקציה מקבלת עירימת פנקיקים המיוצגת על ידי רשימה מספרים לא ממויננת.

השתמשו בfonקציות שיצרתם בסעיפים א' וב' על מנת להפוך את העירימה לעירימה מושלמת - כולם לסדר את עירימת הפנקיקים בסדר יורד כפי שתואר בשאלת.

אין להשתמש בfonקציות מובנות של פיתון או לבצע שינויים מיקום בעירימה שלא באמצעות `fonקציית flip`.



שאלה 2

נרצה לחלק את הפנק'יקים בין סועדים בצורה שווה. נגדיר את החלוקת באופן הבא:

- כל סועד מקבל שני פנק'יקים.
- כל הסועדים יקבלו פנק'יקים שגודלם המוצע שווה.

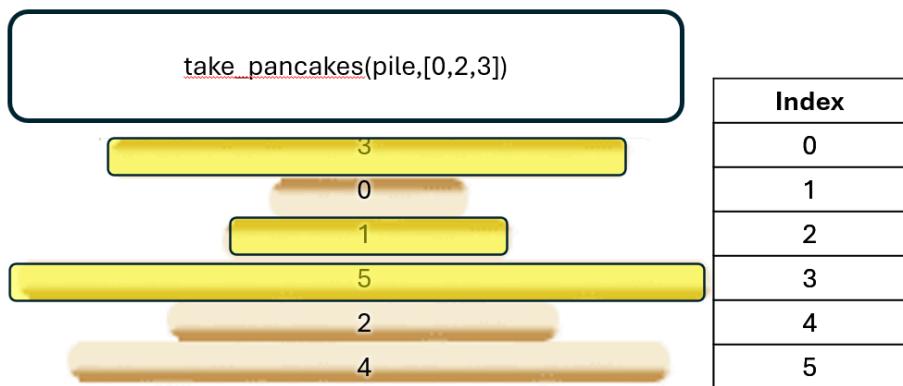
 עuffy א'

ממשו את הפונקציה `take_pancakes(pile,index_list)`

הfonk'icah מקבלת ערימה של פנק'יקים (`pile`) כרשימת מספרים, ורשימה של מספרים טבעיים המיצגים אינדקסים (`index_list`). הפונקציה נעזרת ברשימת האינדקסים (`index_list`), מוציאה את הפנק'יקים הנמצאים במקומות המתאימים מתוך `pile`, ומהזירה רשימה חדשה המכילה את גדי הפנק'יקים שהוזאו.

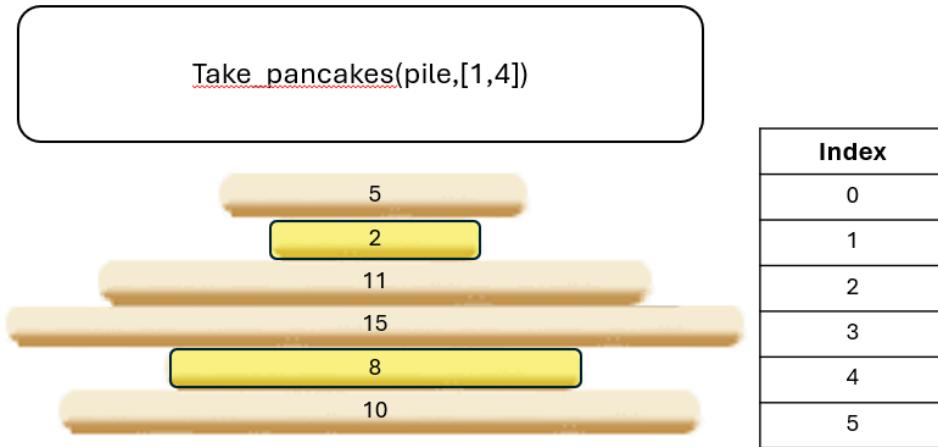
רמז: היעזרו בשיטה `pop` של רשימה

לפניכם דוגמה של שימוש בfonk'icah, בצד שמאל מסומנים הפנק'יקים באינדקסים המתאים לאינדקסים בפרמטר `index_list` והם הפנק'יקים שיילקחו מהערימה כתוצאה מהפעלת הפונק'icah:



הערך שיוחזר: [3,1,5]
הערך לאחר הפעלת הפעולה: [0,2,4]

Take_pancakes(pile,[1,4])



הערך שיוחזר: [2,8]
העימה לאחר הפעלת הפעולה: [5,11,15,10]

סעיף ב'

ממשו את הפונקציה `(pile, pancakes)` המיצגות שתי ערים של פנקיקים כאשר הפונקציה מקבל כקלט שתי רשימות של מספרים טבעיות המיצגות שתי ערים של פנקיקים כאשר ידוע שהעימה `pile` היא ערים מושלמת (מספרית מתקטן לגודל). על הפונקציה להכניס את הפנקיקים מהעימה `pancakes` לעירמה הראשית `pile` תוך שמירה על המון בסדר יורד של העימה הראשית `pile`. יש להכניס כל פנקיק למקום המתאים לו בעימה `pile`. אין **למיין את העימה בתהליך** – פתרון שמנמן את העימה בתהליך יפסל (בין אם בעזרת פונקציה מובנית של פיטון או בעזרת פונקציות שנכתבו בסעיפים קודמים) רמז: היעזרו בשיטה `insert` של רשימה לדוגמה עברו הרשימה

```
pile = [2,3,4,7,12]
```

והפעלת הפעולה:

```
insert_pancakes(pile, [1,6,13])
```

תתקבל התוצאה:

```
[1, 2, 3, 4, 6, 7, 12, 13]
```

סעיף ג'

ממשו את הפונקציה `divide_pancakes(pile, total_size)`

הפונקציה מקבלת כקלט `pile` רשימה של מספרים טבעיות המיצגת עירימת פנקיקים ומספר טבעי `total_size` המיצג את סכום הגודלים אותו נרצה לחלק לכל סוג.

על הפונקציה להחזיר רשימה מקוונת של רשימות **אורקל 2** (כלומר רשימה המכילה רשימות שבכל אחת נמצאים שני פנקיקים בדיאק) כך שכל רשימה מוכלת היא עירימת פנקיקים שסכום הגודלים שלה שווה ל-`total_size`.

לדוגמה עבור הערימה:

```
pile = [2,3,4,4,5,5,6,7,8,9,12]
```

והפעלת הפונקציה עם הפרמטר 10:

```
divide_pancakes(pile,10)
```

יתקבל הפלט הבא:

```
[[7, 3], [8, 2], [6, 4], [5, 5]]
```

עבור הפעלת הפונקציה עם הפרמטר 12:

```
divide_pancakes(pile,12)
```

יתקבל הפלט הבא:

```
[[9, 3], [8, 4], [7, 5]]
```

אין משמעות לסדר של הפנקיקים בפלט.