

מבוא לתכנות מערכות

תרגיל בית 3

סמסטר חורף 2022

תאריך פרסום: 16/01/2022

תאריך הגשה: 27/01/2022

משקל התרגיל: 5% מהציון הסופי (תקף)

מתרגלים אחראים: אורטל כהן

1 הערות כלליות

- שימו לב: לא יינתנו דחיות במועד התרגיל. תכננו את הזמן בהתאם.
- לשאלות בנוגע להבנת התרגיל יש לפנות לסדנאות של אחד מהמתרגלים, או לשאול בפורום של הקורס בפיאצה. לפני שליחת שאלה - נא וודאו שהיא לא נענתה כבר ב-F.A.Q או בפיאצה, ושהתשובה אינה ברורה ממסמך זה, מהדוגמא ומהבדיקות שפורסמו עם התרגיל.
- קראו מסמך זה עד סופו לפני שאתם מתחילים לממש. יתכן שתצטרכו להתאים את המימוש שלכם לחלק עתידי בתרגיל. תכננו את המימוש שלכם לפני שאתם ניגשים לעבוד. רצוי לעבור על הדוגמא שפורסמה לפני תחילת הפתרון.
- כל חומר נלווה לתרגיל נמצא על השרת בתיקייה [~mtm/public/2122a/ex3](https://mtm/public/2122a/ex3).
- חובה להתעדכן בעמוד ה-F.A.Q ובפיאצה - הכתוב שם מחייב.
- העתקות קוד בין סטודנטים תטופלנה בחומרה!
- מומלץ מאוד מאוד לכתוב את הקוד בחלקים קטנים, להעביר כל חלק בנפרד אל השרת, ולבדוק שהוא עובד באמצעות שימוש בטסטים.

2 מתמיקאה – Matamikya

2.1 הקדמה

הטכניון החליט ליצור גרסה נוספת למערכת הממוחשבת ששולטת במלאי המוצרים הקיימים לעסק של מתמיקאה, בהיקף קטן יותר. עליכם לממש מערכת שקולטת קובץ קלט ומבצעת עליו פעולות.

2.2 תיאור קובץ הקלט

מבנה הקובץ שתקבלו יהיה כך:

```
add product <name> <price> <amount>
change amount <name> <new amount>
add product <name> <price> <amount>
add product <name> <price> <amount>
ship order <product name>, <product amount> -- <product name>, <product
amount>
change amount <name> <new amount>
```

כלומר יש בקובץ שלושה סוגים של שורות,

סוג ראשון:

שורות המתחילות בשתי המילים add product, שורות כאלה מתארות מוצר עבור מלאי מסוים.

סוג שני:

שורות המתחילות במילה change amount, שורות כאלה מתארות דרישה לעדכון הכמויות של מוצר מסוים במלאי.

- אם השם של המוצר לא קיים, יש להתעלם משורה זאת.

סוג שלישי:

שורות המתחילות בשתי המילים ship order, בשורות אלו יש מספר לא מוגבל של זוגות מהצורה <product amount>, <product name>, כאשר כל זוג מופרד באמצעות התווים "--". שורה זאת מייצגת הזמנה של מוצרים ומכירתם. יש לעדכן את המלאי בהתאם.

- אם עבור מוצר מסוים הכמות הנדרשת אינה קיימת במלאי (כלומר הכמות לאחר ההזמנה קטנה ממש מ-0), לא יהיה כל שינוי עבור מוצר זה (כלומר מוצר זה לא יימכר). במקרה זה, ההזמנה כן תתבצע ויש להתעלם ממוצר זה.

- אם מוצר מסוים לא קיים במערכת או שכמות המוצר שלילית או שווה לאפס, יש לדלג אך ורק על מוצר זה.

- יכול להיות שאותו מוצר יופיע מספר פעמים בפקודה זאת, במקרה זה אין להתעלם ממוצר זה, אלא לבצע את ההזמנה כרגיל.

פירוש הפרמטרים בשלושת סוגי השורות הוא:

- name - שם המוצר, לא יכולים להיות שני מוצרים שונים בעלי שמות זהים. אם נכנס מוצר חדש בעל שם זהה, התוכנית לא תעדכן את המלאי. ניתן להניח כי מוכנס שם בעל אותיות קטנות באנגלית.
- price – מחיר המוצר. אם price שלילי, אין להוסיף את המוצר למלאי. price אמור להיות מטיפוס float.
- amount – כמות המוצר ההתחלתית במלאי. אם amount שלילי, אין להוסיף את המוצר למלאי. amount אמור להיות מטיפוס float.
- new amount – כמות המלאי שיש לשנות עבור מוצר מסוים. אם הכמות חיובית נוספים מוצרים למלאי, אם הכמות שלילית מוסרים מוצרים מהמלאי. ואם הכמות שווה ל-0, המלאי לא משתנה. יכול להיות שהכמות החדשה של המוצר שלילית. אמור להיות מטיפוס float.
- product name – שם המוצר אותו יש למכור.
- product amount – כמות המוצר אותו יש למכור. אמור להיות מטיפוס float.

הנחות על קובץ הקלט:

- ניתן להניח שמבנה הקובץ תמיד יהיה תקין ושכל המידע בתוכו נכון.
- שמות מוצרים לא יכילו רווחים, ולא יהיו רווחים בתוך ערכי הפרמטרים שבתוך השורות אלא רק בין הפרמטרים.

- אין חשיבות לסדר בין השורות, כלומר יש לבצע את השורות לפי הסדר בקובץ. לדוגמא, אין לבצע את כל השורות מהסוג הראשון בהתחלה.
- ניתן להניח כי בשורות מהסוג השלישי תמיד יהיה את שם המוצר ואחריו את כמותו. בנוסף, ניתן להניח כי כל שני מוצרים יופרדו באמצעות התווים "--".

2.3 מימוש מציאת מוצר הנמכר ביותר

ממשו את הפונקציה הבאה:

`find_best_selling_product(file_name)`

אשר מקבלת שם של קובץ בפורמט קובץ הקלט ומחזירה tuple מהצורה

`(product name, current selling price)`

כאשר `product name` הוא שם המוצר הנמכר ביותר וה-`current selling price` הוא המחיר הכולל של כל ההזמנות שהתרחשו עבור מוצר זה.

הערות:

- אם יש מספר מוצרים נמכרים ביותר, יוחזר המוצר עם השם הנמוך ביותר בסדר לקסיקוגרפי.
- אם קובץ הקלט ריק, או אין מוצרים במלאי. עבור `product name` תוחזר מחרוזת ריקה ועבור `current selling price` יוחזר הערך 0.

ניתן להניח שהפרמטרים אשר הפונקציה תקבל חוקיים. כלומר הפרמטר של שם הקובץ מייצג קובץ קיים עם הרשאות קריאה.

המלצות למימוש:

- לפענוח קובץ הקלט מומלץ להשתמש בפונקציה `split` של פייתון שמפרקת מחרוזת למילים. ניתן לקרוא מידע נוסף על `split` כאן.

דוגמה לשימוש:

```
>>> '1 2 3'.split()
['1', '2', '3']
```

2.4 מימוש מציאת k המוצרים היקרים ביותר

ממשו את הפונקציה הבאה:

`find_k_most_expensive_products(file_name, k)`

פונקציה זאת מקבלת שם של קובץ בפורמט קובץ הקלט, `file_name`, ומספר `k`. פונקציה זאת מחזירה רשימה שמכילה את שמות `k` המוצרים היקרים ביותר בסדר יורד לפי סדר מחירם. אם יש מספר מוצרים יקרים באותה מידה, הם יופיעו ברשימה בסדר עולה לקסיקוגרפית לפי שמם.

הערות:

- ניתן להניח כי k הינו מספר שלם.
- אם k שלילי או שווה לאפס, תוחזר רשימה ריקה.
- אם קובץ הקלט ריק, תוחזר רשימה ריקה.
- אם k גדול ממספר המוצרים הקיימים, יש להדפיס את כל המוצרים לפי הסדר המתואר.

3 עבודה עם מחרוזות

3.1 ממשו את הפונקציה הבאה:

`get_palindrom_dict(str)`

פונקציה זאת מקבלת מחרוזת ומחזירה מילון. המילון יכיל מפתחות שהם מספרים, והמספר t ימפה לרשימה שתכיל את כל תתי המחרוזות של `str` שהם פלינדרומים באורך t . כאשר פלינדרום הוא מחרוזת שקריאתו מימין לשמאל ומשמאל לימין זהה.

לדוגמא, עבור הקלט:

`AbbAcaBBa`

הפונקציה תחזיר את המילון:

```
{  
  1: ['A', 'b', 'b', 'A', 'c', 'a', 'B', 'B', 'a'],  
  2: ['bb', 'BB'],  
  4: ['AbbA', 'aBBa']  
}
```

הערות:

- ניתן להניח כי `str` מטיפוס מחרוזת.
- אם עבור מספר מסוים אין תתי מחרוזות שהן פלינדרומים, מספר זה לא יופיע במילון.
- בשאלה זאת, המחרוזת הריקה אינה נחשבת כפלינדרום. אולם, מחרוזת בעלת תו אחד כן נחשבת כפלינדרום.
- אם אין כלל תתי מחרוזות שהם פלינדרומים, הפונקציה תחזיר מילון ריק.
- יש להבדיל בין אותיות קטנות לגדולות. כלומר התו 'a' שונה מהתו 'A'.

3.2 ממשו את הפונקציה הבאה:

check_match(str)

בשאלה זאת נגדיר **החלפה** בין **מחרוזת אחת למחרוזת שנייה**. **מחרוזת אחת** יכול להתחלף **למחרוזת שנייה** אם כל המופעים של תו במחרוזת הראשונה יכולים להתחלף עם תו אחר יחיד בלבד **במחרוזת השנייה**, כאשר נשמר סדר התווים **לאחר ההחלפה**.

דוגמאות למחרוזות **שיכולות להתחלף**:

“door”, “keep”

“xatal”, “tpmpd”

דוגמאות למחרוזות **שהמחרוזת השמאלית לא יכולה להתחלף עם הימנית**:

“dad”, “sad”

שימו לב כי למרות ש-“dad” אינו יכול להתחלף ל-“sad”, “sad” כן יכול להתחלף ל-“dad”.

(מכיוון שהמופע של d של המחרוזת הימנית מתחלף עם התווים d-i s במחרוזת השמאלית).

“check”, “help”

(התווים באורכים שונים).

הפונקציה check_match מקבלת מחרוזת ומחזירה True, אם המחרוזת המורכבת מהתווים במקומות האי-זוגיים של str **(המחרוזת הראשונה בהגדרה שלעיל)** יכולה להתחלף למחרוזת המורכבת במקומות הזוגיים של str **(המחרוזת השנייה בהגדרה שלעיל)**. אחרת, הפונקציה מחזירה False.

הנחות:

- ניתן להניח כי str מטיפוס מחרוזת.
- אם המחרוזת ריקה יש להחזיר True.
- בתרגיל זה המחרוזת מתחילה מאינדקס 1.

4 הגשה

אתם נדרשים להגיש רק שני דברים:

- קובץ hw3_part1.py (בשם זה) שמכיל את הקוד שכתבתם ואת הפונקציות של חלק 2
find_best_selling_product - ו find_k_most_expensive_products
- קובץ hw3_part2.py (בשם זה) שמכיל את הקוד שכתבתם ואת הפונקציות של חלק 3
get_palindrom_dict - ו check_match

שימו לב, אי עמידה במבנה הגשה זה תגרור הורדת נקודות

מסופק לכם תסריט בדיקה של מבנה ההגשה שנקרא final_check בתיקיית התרגיל
בשרת, ע"מ לעשות בדיקה זו העבירו את קובץ zip שלכם לשרת ותבצעו את הבדיקה
הבאה:

hw3.zip ./final_check

התסריט יבדוק שהיררכיית הקבצים נכונה, ובפרט שמבנה הקובץ שאתם מגישים נכון ויעביר
טסט פשוט וקוד שכתבתם, **טסט זה אינו מספיק!**

הקפידו לבדוק את הקוד שלכם עם מגוון טסטים, אנו ממליצים גם על שיתוף טסטים בין
הסטודנטים.