מבוא למחשב – שפת פייתון (234128)

מועד א' סמסטר אביב תש"ף

02 באוגוסט 2020

משך המבחן: 119 דקות. אין יציאה לשירותים!
חומר עזר: אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני.

הנחיות והוראות:

1. יש לכתוב פתרונות בכתב-יד באופן ברור, נקי ומסודר. יש להתחיל את הפתרון לכל שאלה **בעמוד חדש**!
2. בכל עמוד חדש יש לכתוב את מספר תעודת הסטודנט.
3. אין לכתוב תשובות **בעפרון**. אין לכתוב **באדום.**
4. מותר להשתמש בפונקציות שמופיעות **בטבלה** בעמוד הבא ובפונקציות קלט/פלט ()print, input().
5. **אין להשתמש ללא מימוש בפונקציות מובנות של פייתון** (למעט הפונקציות בטבלה) **או בפונקציות שמומשו בכיתה,** אלא אם צוין אחרת במפורש בשאלה.
6. בכל השאלות, הנכם רשאים להגדיר (לממש) פונקציות עזר כרצונכם, אלא אם צוין אחרת במפורש בשאלה. סדר כתיבת הפונקציות איננו חשוב.
7. בכל שאלה ניתן להשתמש בפונקציות המוגדרות בסעיפים קודמים, גם אם לא פתרתם סעיפים אלו, אלא אם נדרש אחרת.
8. אין להשתמש במשתנים גלובליים ו/או סטטיים.
9. אין להשתמש במילונים (dict).
10. יש לעמוד בעקרונות תכנות שנלמדו בקורס. **שכפול קוד הוא סיבה להורדת ציון**.
11. אין צורך לבדוק תקינות קלט אלא אם צוין אחרת במפורש בשאלה.
12. נוהל **"לא יודע":** אם התשובה לסעיף מסוים (או שאלה מסוימת) אינה ידועה, הנכם ראשים לכתוב בסעיף זה "**לא יודע/ת**" בצורה ברורה (!) ולקבל 20% מהניקוד לאותו סעיף.

צוות הקורס 234128
מרצים: ד"ר ילנה נופברי
מתרגלים: רוג'ה חאיק, אוהד ברטה, יותם אמיתי, אביגיל כהן רמון, מוחמד נאסר, יונתן אלול, תום אברך, וסים מח'ול.

בהצלחה!

רשימת פונקציות ופעולות :

פעולה	סיבוכיות זמן
my_list.append(x) / my_list += [x]	O(1)
my_list.extend(list2) / my_list += list2	O(k), k=len(list2)
new_list=my_list[start: stop: step]	O(m), m=len(new_list)
len(my_list)	O(1)
my_list[i]	O(1)
sum(my_list)	O(n)
min(my_list) / max(my_list)	O(n)
x in my_list	O(n)
range(start, stop, step)	O(m), m is the length of range
enumerate(my_list)	O(n)
int(x) / round(x) / bool(x)	O(1)
abs(x)	O(1)
ord(c) / chr(x)	O(1)
my_list.sort()	O(n log(n))

שאלה 1 (30 נק') **יש לכתוב את הפתרון בעמוד חדש ולסמן את מספר הסטודנט**

סעיף א' (25 נק')

לקוח רוצה לקנות בחנות 2 מוצרים שונים. ללקוח יש סכום כסף נתון – money,
והוא רוצה לבזבז את כל כספו בדיוק.

עליכם לממש את הפונקציה

```
def buy_2_items(lst, money)
```

אשר מקבלת את רשימת המחירים של מוצרים בחנות `lst`, ואת סכום הכסף שיש ללקוח `money`.
הפונקציה מחזירה True אם קיימים בדיוק 2 מוצרים שסכום המחירים שלהם שווה ל-`money` ,
או False אחרת.

דוגמאות:

- בהינתן הרשימה `lst=[14 ,32 ,76 ,89 ,22 ,19 ,1]` עבור הסכום money=100,
הפונקציה מחזירה False.
- עבור אותה הרשימה וסכום הכסף 108 הפונקציה מחזירה True כיוון 108=32+76.
- גם עבור money=54 הפונקציה מחזירה True כי 54=32+22.

דרישת סיבוכיות: על הפונקציה לעמוד בסיבוכיות $O(n\ Log\ (n))$, כאשר n אורך הרשימה `lst`.

הערות:

- פתרון פחות יעיל יקבל ניקוד חלקי.
- הרשימה כוללת לפחות 2 איברים, וכל האיברים ברשימה שונים זה מזה.
- ניתן לשנות את הרשימה.

סעיף ב' (5 נק')

עליך להסביר בקצרה את סיבוכיות זמן של הפתרון שלך: יש להסביר על סיבוכיות של כל חלק של הפתרון,
ולסכם לסיבוכיות הפונקציה `buy_2_items` כולה.

```
def buy_2_items(lst, money):
    lst.sort()
    start = 0
    end = len(lst) - 1
    while start < end:
        if (lst[start] + lst[end]) == money:
            return True
        if (lst[start] + lst[end]) < money:
            start += 1
        else:
            end -= 1
    return False
```

שאלה 2 (35 נק') **יש לכתוב את הפתרון בעמוד חדש**

למועדון "השם המוזר" יש חוקים מאוד נוקשים לגבי שמות של מי רשאי להצטרף למועדון ומי לא.
כדי להתקבל, דרוש כי השם של המועמד יעמוד בקריטריונים הבאים:

1. השם באנגלית חייב להכיל את האותיות המרכיבות את המילה muzar (לפחות פעם אחת כל אות),
2. מספר האותיות של שם המועמד שנמצאים במקומות האי-זוגיים באלף-בית (כאשר האות 'a' נמצאת במקום אפס), אסור לו להיות גדול מ-5,
3. אסור לאות הראשונה להיות זהה לאות האחרונה.

סעיף א' (5 נק')

עליכם לממש את הפונקציה `def abc_index(letter)`
אשר מקבלת אות קטנה `letter` ומחזירה את המרחק בינה לבין האות `'a'`.

דוגמאות:

- עבור `'a'` הפונקציה מחזירה 0,
- עבור `'b'` הפונקציה מחזירה 1, ..., עבור `'z'` הפונקציה מחזירה 25.

```
def abc_index(letter):
    return ord(letter) - ord('a')
```

סעיף ב' (15 נק')

עליכם לממש את הפונקציה `def make_histogram(name)`
אשר מקבלת מחרוזת `name` של שם המועמד ומחזירה רשימה של שכיחות האותיות שמרכיבות את השם.

דרישת סיבוכיות זמן: חייבת להיות לינארית. פתרון פחות יעיל **לא מתקבל**.

דוגמה:

- עבור השם "mama zubar" השכיחות של `'a'` שווה 3, השכיחות של `'m'` שווה 2 וכדומה לשאר האותיות.

הערה:

- **חובה** להשתמש בפונקציות מסעיף **א'** אפילו אם לא פתרתם אותו.
- **אסור** להשתמש בפונקציות מובנות או פעולות פייתון על מחרוזות.
- **אין להשתמש במילונים (dict).**
- שם המועמד כולל רק אותיות קטנות באנגלית. יש להתעלם מרווחים בתוך שם המועמד.

```
def make_histogram(name):
    histogram = [0] * ((ord('z') - ord('a')) + 1)
    for s in name:
        if ord('z') >= ord(s) >= ord('a'):
            histogram[abc_index(s)] += 1
    return histogram
```

סעיף ג' (15 נק') **יש לכתוב את הפתרון בעמוד חדש**

עליכם לממש את הפונקציה `def check_weird_name(name)`
אשר מקבלת מחרוזת `name` של שם המועמד ומחזירה True אם השם עומד בכל הקריטריונים (1-2-3),
או False אחרת.

דרישות:

- **סיבוכיות זמן**: חייבת להיות לינארית ביחס לאורך המחרוזת `name`. הפתרון פחות יעיל **לא מתקבל**.

הערה:

- **חובה** להשתמש בפונקציות מסעיף **א'** ומסעיף **ב'** אפילו אם לא פתרתם אותם.
- **אין להשתמש במילונים (dict).**
- שם המועמד כולל רק אותיות קטנות באנגלית. יש להתעלם מרווחים בתוך שם המועמד.
- אפס הוא מספר זוגי.

דוגמאות:

- עבור השם "mama zubar" הפונקציה מחזירה True.
- עבור "avneri luzum" הפונקציה מחזירה True.
- עבור "rami zubar" הפונקציה מחזירה False (אות ראשונה זהה לאות אחרונה).
- עבור "shoshke rednose" הפונקציה מחזירה False (לא מכיל את האותיות של muzar).
- עבור "avneri ben luzum" הפונקציה מחזירה False: מספר האותיות במקומות האי-זוגיים באלף-בית גדול מ-5.

```
def check_weird_name(name):
    histogram = make_histogram (name)
    if str[0] != str[-1]:
        if sum(histogram[1::2]) <= 5:
            if histogram[abc_index('m')] and histogram[abc_index('u')] and\
              histogram[abc_index('z')] and\
              histogram[abc_index('a')] and histogram[abc_index('r')]:
                return True
    return False
```

שאלה 3 (35 נק') **יש לכתוב את הפתרון בעמוד חדש**

הגדרה: בהינתן מטריצה ריבועית `mat` בעלת N שורות ו-N עמודות **(N מספר זוגי)**,
מטריצה "**מוחלפ-לפת**" היא מטריצה שהוחלפו לה סדרי האיברים בשלושה שלבים הבאים:
(1) **קודם כל**, יש להחליף כל שורה באינדקס **זוגי** עם השורה הבאה שנמצאת באנדקס **אי-זוגי**.
(2) **אחר כך**, יש להחליף כל עמודה באינקס **זוגי** עם העמודה הבאה שנמצאת באנדקס **אי-זוגי**.
(3) **ואחר כך**, יש להפוך את סדר הופעת איברי האלכסון הראשי בצורה סימטרית (ר' דוגמה).

עליכם לממש את הפונקציה `def switch_witch_matrix(mat)`
אשר מקבלת מטריצה ריבועית `mat` ומחזירה את המטריצה המוחלפ-לפת.

דוגמה לקריאה לפונקציה:

```
mat=[[1,2,3,4], [5,6,7,8], [9,10,11,12], [13,14,15,16]]
print(switch_witch_matrix(mat))
```

סדר הפתרון: יש לפתור את השאלה בשלושה שלבים כפי שמתואר בהגדרה.

switch rows: כך המטריצה mat נראית אחרי החלפת שורות בהתאם ל- (1)
switch columns: כך המטריצה (שכבר הוחלפו לה שורות) נראית אחרי החלפת עמודות בהתאם ל- (2)
switch main diagonal: כך נראית המטריצה אחרי החלפת איברי האלכסון הראשי בהתאם ל- (3)

mat:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Switch rows:

5	6	7	8
1	2	3	4
13	14	15	16
9	10	11	12

Switch columns:

6	5	8	7
2	1	4	3
14	13	16	15
10	9	12	11

Switch main diagonal:

11	5	8	7
2	16	4	3
14	13	1	15
10	9	12	6

```
def switch_witch_matrix(mat):
    #part I
    even_rows = mat[::2]
    odd_rows = mat[1::2]
    mat[::2] = odd_rows
    mat[1::2] = even_rows

    #part II
    for i,row in enumerate(mat):
        even = row[::2]
        odd = row[1::2]
        row[::2] = odd
        row[1::2] = even
        mat[i] = row

    #part III
    diag = [mat[i][i] for i in range(len(mat))]
    reversed_diag = diag[::-1]

    for i in range(len(mat)):
        mat[i][i] = reversed_diag[i]
    return mat
```