

**אלגוריתמים במולטימדיה ולמידת מכונה בסביבת פייתון**

מס' זהות:

הוראות:

1. המבחן נעשה במחשב.
2. אין שימוש בחומר עזר.
3. מותר להשתמש במחשבון, דף ריק מתוכן וכלי כתיבה.
4. משך הבחינה - 3 שעות
5. במבחן שלוש חלקים:
  - חלק ראשון חובה - יש לענות על כל השאלות.
  - חלק שני בחירה - יש לענות על שתי שאלות מתוך שלוש.
  - חלק שלישי בונים - לא חובה לענות על חלק זה.
6. אין לעזוב את המבחן לפני שהוגש כנדרש (אחרת הציון 0 אוטומטי)
7. שימו לב! לא ליצור את הפרויקט בקונן C (אם קיים קונן נוסף) אחרת יש סיכוי לעובדן מידע.
8. שימו לב! תוודאו שאתם עובדים בגרסת פייתון נכונה.
9. יש ליצור קובץ פונקציות אחד בלבד ובו לממש את כל הפונקציות הנדרשות.
10. כל פונקציה שאתם מתבקשים לכתוב חייבת להיות בעלת שם זהה לזה שבשאלה! כולל התייחסות לאותיות קטנות/גדולות. פונקציה שהשם שלה לא יהיה רשום במדויק, לא תקבל ניקוד.
11. אין להגיש שום סקריפט שלא התבקשתם לכתוב וזה כולל סקריפט הבדיקות שלכם.
12. כל קטע קוד שלא קשור לאחת הפונקציות שהתבקשתם לכתוב יגרור הורדה בציון וזה כולל פלטים שלא התבקשתם לעשות.
13. שימו לב! יש להתחשב במקרי הקצה שעלולים להיות. לדוגמה קלט לא תקין או כתובת שלא קיימת. במקרה של שגיאה יש להחזיר None
14. אתם יכולים להשתמש בכל פונקציה שתמצאו לפתרון המבחן.
15. במהלך המבחן אין גישה לאינטרנט.
16. מי שמסיים את המבחן מחכה במקומו עד להגעת אוסף הבחינות בכדי להגיש את הקוד.
17. יש לרשום את המספר ת.ז. על גבי טופס זה ולהגיש אותו בסוף המבחן.
18. את הסקריפט שהנכם מגישים יש לקרוא בשם המתחיל באות "s" קטנה וללא רווח מס' הת.ז. שלכם. לדוגמה: s123456789
19. במידה וישנה פונקציה שאינכם יודעים מה יעודה, נסאו פקודה:  
print FunName.\_\_doc\_\_

**פונקציות עזר אפשריות**

**Basic functions:**

isinstance, id, hex, ord, str, eval, enumerate, list, range, raw\_input, tuple, type, zip, chr, cmp, in, not, or, and; **String operations:** count, endswith, find, isdecimal, isdigit, isnumeric, join, lower, upper, replace, split; **List operations:** append, extend, insert, remove, pop, count; **Dictionary operations:** has\_key, items, keys, values; **Set operations:** issubset, union, intersection, difference, symmetric\_difference, copy, add, remove, discard, pop

**copy:**

copy

**os:**

getcwd, listdir, path.isfile, path.exists, path.join, path.isdir

**numbers:**

Number

**numpy:**

zeros, ones, eye, mean, median, sum, prod, dot, transpose, trace, ceil, floor, round, max, min, argmax, argmin, sort, argsort, reshape, concatenate, all, any, pi, asarray, copy, logical\_or, logical\_and, logical\_not, logical\_xor, array\_equal, isnan, linspace, arrange, mod, exp, log, sin, cos, tan, sinc, sinh, cosh, tanh, arcsin, arccos, arctan, arctan2, arcsinh, arccosh, arctanh, intp, int8, int16, int32, int64, uint8, uint16, uint32, uint64, float16, float32, float64, complex64, complex128, random.rand, random.randn, random.randint, random.random, random.shuffle, random.normal, random.uniform, ndarray.astype, linalg.eig, linalg.eigh, linalg.eigvals, linalg.norm, linalg.det, linalg.matrix\_rank

**cv2:**

copyMakeBorder, BORDER\_REPLICATE, BORDER\_CONSTANT, BORDER\_REFLECT, filter2D, split, merge, cvtColor, COLOR\_BGR2GRAY, COLOR\_GRAY2RGB, imread

**matplotlib.pyplot:**

figure, subplot, subplot2grid, plot, stem, scatter, show, bar, hist, hist2d, pie, psd, imshow, axis, grid close, colorbar, draw, gca, gcf, legend, loglog, semilogx, semiology, pause, title, xlabel, ylabel, xlim, ylim, xticks, yticks

חלק א' (חובה): 50%

**שאלה 1 (5 נק.).**

1. הגדירו פונקציה בשם: **myPlusMatrix**  
הפונקציה צריכה לקבל מספר טבעי  $n$ .  
הפונקציה צריכה להחזיר מטריצה ריבועית (מערך דו-מימדי) ממימד  $2n+1$  בה יש אחדים בעמודה ובשורה האמצעיות.

דוגמה:

$n = 1$   
 $\text{myMat} = \text{myPlusMatrix}(n)$

תוצר הפונקציה יהיה:

$$\text{myMat} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

**שאלה 2 (10 נק.).**

2. הגדירו פונקציה בשם: **countMyStrings**  
הפונקציה צריכה לקבל מחרוזת.  
הפונקציה צריכה להחזיר מילון המכיל את כל המילים הקיימים במחרוזת ללא חזרות וללא חשיבות לאותיות גדולות/קטנות בתור המפתחות של המילון וערך עבורן יהיה מספר הופעות של כל אחת מהמילים במחרוזת שהתקבלה, כולל מקרים בהם מילה היא חלק ממילה אחרת.

לדוגמה:

$\text{myStr} = \text{'Hello Hello We Do Hi no No no we Donot Hey HE he heY'}$   
 $\text{myDict} = \text{countMyWords}(\text{myStr})$

תוצר הפונקציה יהיה:

**myDict :**

hello : 2,      we : 2,      do : 2,      hi : 1  
no : 4,      donot : 1,      hey : 2,      he : 6

**שאלה 3 (10 נק.).**

3. הגדירו פונקציה בשם: **myListMean**  
הפונקציה מקבלת רשימה המכילה מספרים ורשימות נוספות במבנה דומה לרשימה עצמה.  
הפונקציה צריכה להחזיר ממוצע של כל האיברים הנמצאים ברשימה שהתקבלה כולל כל תתי הרשימות.

לדוגמה:

$\text{myList} = [1, [2], 3, [1, -1, [2, 3, [1.0], 1], 2], -3, 1.0, [[0]]]$   
 $\text{myMean} = \text{myListMean}(\text{myList})$

תוצר הפונקציה יהיה:

$\text{myMean} = 1.0$

**שאלה 4 (10 נק.)**

4. הגדירו פונקציה בשם **UpFolderConatin**:

הפונקציה לא מקבלת שום קלט.

הפונקציה צריכה להחזיר רשימה של שמות כל הקבצים והתיקיות הנמצאים בתיקיה בה נמצאת תיקיית העבודה של הפרויקט בו אתם עובדים.

לדוגמה:

```
myList = UpFolderConatin()
```

**שאלה 5 (15 נק.)**

5. הגדירו פונקציה בשם **myTwoLiner**:

```
def TwoLiner(img):
    if img.__class__ != np.ndarray:
        return None

    im = img.copy()
    D = im.shape
    th = np.int64(np.round(D[1] / 5.))
    wi = np.int64(np.round(D[1] / 30.))
    if len(D) != 3 or D[0] < 30:
        return None

    # -----
    for m in range(th - wi, th + wi + 1):
        for n in range(D[0]):
            im[n, m, 0] = 150
            im[n, m, 1] = 255
            im[n, m, 2] = 0
        for m in range(4*th - wi, 4*th + wi + 1):
            for n in range(D[0]):
                im[n, m, 0] = 150
                im[n, m, 1] = 255
                im[n, m, 2] = 0
    # -----

    return im
```

תחליפו את קטע הקוד שבין הקווים השבורים בשורת קוד אחת בלבד.



## מכון טכנולוגי חולון Holon Institute of Technology

חלק ב' (בחירה): 50%

### שאלה 6 (25 נק.)

6. הגדירו פונקציה בשם: **myLinearDepandacyRemoval**  
הפונקציה צריכה לקבל מטריצת ndarray של נתונים בייצוג float64.  
הפונקציה צריכה לחשב ממוצע בין כל השורות שבמטריצה ולהחסיר אותה מכל שורה ושורה.  
לאחר מכן יש לחשב את מטריצת שונות המשותפת של המטריצה המתקבלת.  
למטריצת שונות המשותפת יש לחשב ערכים עצמיים ווקטורים עצמיים.  
יש להעתיק את מטריצת הנתונים למרחב פורש בעל שונות מקסימלית וללא תלות לינארית  
(להיפתר מוקטורים עצמיים של ערכים העצמיים ששווים ל 0). תניחו שכל ערך הקטן מ  $10^{-9}$   
הוא שווה ל 0.  
תזכורת:

חישוב מטריצת שונות משותפת:

$$CovMat = A^t A \quad \text{כאשר מטריצה } A \text{ מסדר } m \times n : A_{(m,n)}$$

חישוב ערכים עצמיים:

$$C \cdot \bar{u} = \lambda \bar{u}$$

$$U = [\bar{u}_1, \dots, \bar{u}_n]$$

העתקה לינארית:

$$F = A \cdot V$$

דוגמה:

Field = **myLinearDepandacyRemoval**(DataMat)

### שאלה 7 (25 נק.)

7. הגדירו פונקציה בשם: **myGradDescent**  
הפונקציה צריכה לקבל אות וזמן אשר מוגדרים על ידי מערכים של ndarray, נקודת התחלה  
שהיא ערך של זמן, מספר איטרציות וגודל צעד.  
הפונקציה צריכה להחזיר את ערך הזמן אליו הגיע האלגוריתם לאחר כמות האיטרציות  
שניתנה עם גודל צעד הנתון.  
יש להשתמש בגרדיאנט רובסטי. משמע, נגזרת מסדר ראשון אשר פחות רגישה לרעשים.  
דוגמה:

$t_1 = \text{myGradDescent}(\text{signal}, \text{time}, t_0, \text{iterNum}, \text{stepSizr})$

### שאלה 8 (25 נק.)

8. הגדירו פונקציה בשם: **mySignalFiltering**  
הפונקציה צריכה לקבל אות מוגדר על ידי מערך ndarray, מספר טבעי (סדר המסנן) ומספר  
נוסף (T) שערכו אפס או אחד.  
הפונקציה צריכה להחזיר את האות אשר עבר סינון על ידי מסנן MAF סימטרי אם מספר T  
שווה לאפס או על ידי Median Filter אם מספר T שווה לאחד. עבור כל ערך T אחר יש  
להחזיר None. עוצמת הסינון צריכה להיות בהתאם לסדר המסנן.  
יש לעשות ריפוד "המשכה". ריפוד אשר משכפל את האיבר שבקצה.  
דוגמה:

Sig = **mySignalFiltering**(signal, order, T)

חלק ג' (בונוס): 15%

שאלה 9 (15 נק.)

9. הגדירו פונקציה בשם: **myOddIndexSuppression** להלן הפונקציה:

```
def myOddIndexSuppression(img):
```

```
    myImg = img.copy()
```

```
    # -----
```

```
    for m in range(img.shape[0]):
```

```
        for n in range(img.shape[1]):
```

```
            if not np.mod(m+n,2):
```

```
                myImg[m, n] = 0
```

```
    # -----
```

```
    return myImg
```

תחליפו את קטע הקוד שבין הקווים השבורים בשורת קוד אחת בלבד.

**בהצלחה!**