

**מבחן בקורס Big Data, למידת מכונה ומולטימדיה בסביבת**  
**Python**

**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 אוטומטית)
7. שימו לב! יש ליצור את הפרויקט בקונן D אחרת יש סיכוי לעובדן מידע.
8. שימו לב! תוודאו שעתם 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 הודעה בציון.
13. שימו לב! יש להתחשב במקרי הקצה שעלולים להיות. לדוגמה קלט לא תקין או כתובת שלא קיימת.
14. אתם יכולים להשתמש בכל פונקציה שתמצאו לפתרון המבחן.
15. במהלך המבחן אין גישה לאינטרנט.
16. בסוף המבחן תקבלו גישה לאינטרנט למספר דקות בכדי להגיש את הסקריפט לתיבת ההגשה שתהיה פתוחה במודל.
17. מי שמסיים את המבחן לפני הזמן מחכה במקומו עד להגעת המרצה.



## מכון טכנולוגי חולון Holon Institute of Technology

חלק א' (חובה): 50%

### שאלה 1 (5 נק.)

1. הגדירו פונקציה בשם: `myTensTensor`  
הפונקציה צריכה לקבל משתנה יחיד מסוג tuple שמכיל מספרים טבעיים (כמות לא מוגבלת).  
הפונקציה צריכה להחזיר מערך n-מימדי שכל איבר בו יהיה שווה ל 10.

דוגמה:

```
myDim = (3,5)  
myMat = myTensTensor(myDim)
```

תוצר הפונקציה תהיה מטריצה 3x5:

$$\text{myMat} = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$$

### שאלה 2 (15 נק.)

2. הגדירו פונקציה בשם: `myFilesList`  
הפונקציה צריכה לקבל לכל היותר משתנה אחד שהוא הכתובת של תיקיה כלשהי במחשב. משתנה זה חייב להיות מסוג string.  
הפונקציה צריכה להחזיר את רשימת הקבצים (תתי תיקיות לא נחשבות כקבצים) שנמצאים בתיקיה שכתובתה קיבלה הפונקציה.  
במידה והפונקציה לא קיבלה אף משתנה או קיבלה כתובת שלא קיימת במחשב, הפונקציה תחזיר את רשימת הקבצים בתיקית העבודה הנוכחית.

דוגמה:

```
myFolder = 'C:\Users'  
noFolder = 'Hello World'  
list1 = myFilesList(myFolder)  
list2 = myFilesList()  
list3 = myFilesList(noFolder)
```

לקלט זה, רשימות 2 ו 3 יהיו זהות והתוכן שלהן יהיה רשימת הקבצים בתיקית העבודה הנוכחית. רשימה 1 תכיל את רשימת הקבצים הנמצאים בתיקית הנתונה.

פונקציות עזר:

```
os.getcwd()    os.path.exists(myPath)    os.listdir(myPath)  
os.path.isfile(myFilePath)    os.path.join(myPath, fileName)
```

**שאלה 3 (10 נק.)**

3. הגדירו פונקציה בשם: **myImageMedian** הפונקציה צריכה לקבל תמונה. תמונה זו חייבת להיות מסוג ndarray. הפונקציה צריכה להחזיר את ערך החציון של התמונה כ float64. (ערך החציון של כלל האיברים הנמצאים בתמונה ללא חשיבות למימדיה).

דוגמה:

```
myMed = myImageMedian(img)
```

ספריית עזר: numpy

**שאלה 4 (5 נק.)**

4. הגדירו פונקציה בשם: **myImageNegative** הפונקציה צריכה לקבל תמונה. תמונה זו חייבת להיות מסוג ndarray. הפונקציה צריכה להחזיר תמונת שכל ערך בה הוא המשלים ל 255 של התמונה המקורית. זאת אומרת שבכל מקום שבו בתמונה המקורית יש ערך 255, בתמונת תוצר במקום המתאים צריך להיות ערך 0, כל ערך של 254 בתמונה המקורית, מוחלף ב 1 בתמונת התוצר וכך הלאה. אם היינו מחברים את שתי התמונות, היינו מקבלים תמונה שכל הערכים בה הם 255.

דוגמה:

```
newImg = myImageNegative(img)
```

**שאלה 5 (15 נק.)**

5. הגדירו פונקציה בשם: **myFastToneReplacement** להלן הפונקציה:

```
def myFastToneReplacement(img, fromA, toB, equalsC):
    myImg = img.copy()
    # -----
    D = myImg.shape
    if len(D) == 2:
        for i in range(D[0]):
            for j in range(D[1]):
                if myImg[i, j] >= fromA and myImg[i, j] <= toB:
                    myImg[i, j] = equalsC
    elif len(D) == 3:
        for m in range(D[0]):
            for n in range(D[1]):
                for k in range(D[2]):
                    if myImg[m, n, k] >= fromA and myImg[m, n, k] <= toB:
                        myImg[m, n, k] = equalsC
    else:
        return None
    # -----
    return myImg
```

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

```
numpy.logical_and(boolMat1, boolMat2)
```

חלק ב' (בחירה): 50%

**שאלה 6 (25 נק.)**

6. הגדירו פונקציה בשם: **myRGBsplit**  
הפונקציה צריכה לקבל תמונה. תמונה זו חייבת להיות מסוג `ndarray`.  
הפונקציה צריכה לפצל את התמונה למטריצות `R`, `G` ו-`B` וכל אחד מהמטריצות להפוך לתמונה צבעונית שמתארת את הגוונים של הצבה המתאים. זאת אומרת ש-`R` תהיה מטריצת גווני אדום, `G` תהיה מטריצת גווני ירוק ו-`B` תהיה מטריצת גווני כחול.  
השתמש בספריית `matplotlib` בשביל לפלוט את שלושת התמונות אחת ליד השניה עם כיתוב מעל כל תמונה שמציין איזו מטריצה הוא פולט.

פונקציות עזר:

```
plt.title('myString')    plt.subplot(myNumber)    plt.show()
cv2.split(colorImage)    cv2.merge(myTuple)
```

**שאלה 7 (25 נק.)**

7. הגדירו פונקציה בשם: **myMedianFilt**  
הפונקציה צריכה לקבל תמונה בגווני אפור וסדר הפילטר. תמונה חייבת להיות מסוג `ndarray`. סדר הפילטר חייב להיות מספר טבעי.  
הפונקציה צריכה להחזיר את התמונה המפולטרת על ידי מסנן החציון.  
גודל התמונה המתקבלת חייב להיות זהה לזה של התמונה המקורית. הקצבות של התמונה צריכים להיות מוגדרים נכון.  
תזכורת:

$$window = order * 2 + 1$$

דוגמה:

```
medImg = myMedianFilt(img, filtOrder)
```

פונקציות עזר:

```
np.reshape(myMat, myShapeTuple)    np.sort(myArr)
np.median(myMat)    np.zeros(myShapeTuple)    cv2.imread(myStr, 0)
cv2.copyMakeBorder(myMat, top, bottom, left, right, cv2.BORDER_REPLICATE)
```

**שאלה 8 (25 נק.)**

8. הגדירו פונקציה בשם: **myMaskKernel**  
הפונקציה צריכה לקבל תמונה בגווני אפור ומסכה. שתי המטריצות חייבות להיות מסוג `ndarray`.  
הפונקציה צריכה להחזיר את המטריצה המתקבלת כתוצאה מתהליך המיסוך.  
ניתן להניח שמימדים של המסכה הם אי זוגיים (אך לא בהכרח שווים).

דוגמה:

```
maskedImg = myMaskKernel(img, maskKernel)
```

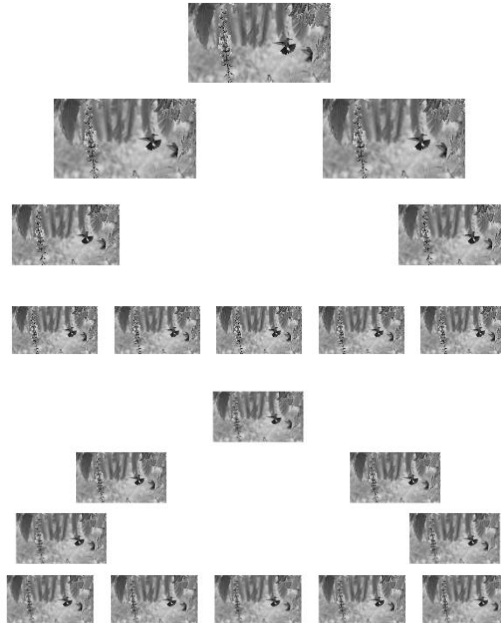
פונקציות עזר:

```
cv2.copyMakeBorder(myMat, top, bottom, left, right, cv2.BORDER_REPLICATE)
np.zeros(myShapeTuple)    cv2.imread(myStr, 0)    np.sum(myMat)
```

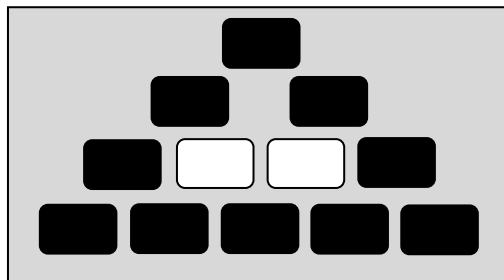
חלק ג' (בונוס): 15%

שאלה 9 (15 נק.).

9. הגדירו פונקציה בשם: `myPlotShape`  
הפונקציה צריכה לקבל תמונה בגווי אפור. תמונה חייבת להיות מסוג `ndarray`.  
הפונקציה צריכה לפלוט גרף של ספריית `matplotlib` שנראה כך:



סקיצה:



שימו לב ששתי התמונות לדוגמה בעצם אותה התמונה. יש הבדלים בתצוגה בגלל גודל החלון ולכן תצוגה יכולה להיות שונה במסכים או מחשבים שונים.  
סקיצה מתארת לכם איך החלונות צריכים להיות ממוקמים בתוך החלון, כאשר שחור זו תמונה ולבן זה מקום ריק מתמונה.

**בהצלחה!**