

### הנחיות

1. במבחן 4 שאלות.
2. פתרון כל השאלות יהיה בסביבת ה-vpl, נתונה לכם החתימה של חלק מהפונקציות לשאלות השונות. **אין לשנות את שמות החתימות או את מספר הפרמטרים בקלט**. מותר לכם להוסיף פונקציות עזר נוספות.
3. בדיקות שאתם רוצים לבצע יש לכתוב **אך ורק** בפונקציית ה-main ולא בשום מקום אחר.

### דגשים

4. אסור לכתוב קוד שלא בתוך הפונקציה. במידה ויהיה קוד כזה, ירדו 5 נקודות מהציון הסופי של הבחינה.
5. הקוד צריך לעבוד ולרוץ. קוד שאינו רץ יקבל מקסימום 75% מניקוד השאלה. יחד עם זאת, קוד שעובד מושלם אינו מבטיח 100% מניקוד השאלה שכן בציון נלקחים בחשבון בהירות ויעילות, ולכן יש להקפיד על כתיבה תקינה וברורה כפי שנלמד בקורס.
6. מאחר ומותר להוסיף פונקציות עזר שאינן נתונות בשלד המבחן.
  - o היה וזו פונקציית עזר לתוכנית הראשית, בתהליך הבדיקות שלכם, יש לזמן את פונקציית העזר מה-main בלבד.
  - o היה וזו פונקציית עזר לפתרון שאלה, יש לזמן את הפונקציה בקוד הפתרון של השאלה.

### **שאלה 1 (20 נקודות):**

ממשו את הפונקציה **הרקורסיבית max\_couple** המקבלת רשימה של מספרים וגודל הרשימה ומחזירה את הסכום **המקסימלי** של זוגות המספרים במקומות **הנגדיים** ברשימה.

אם אורך הרשימה הוא אי זוגי האיבר האמצעי נחשב זוג עם ערכו בלבד.  
 לדוגמא עבור הרשימה [6, 5, 2, 6, 1, 3, 8, 9] הפונקציה תחזיר 13  
 לדוגמא עבור הרשימה [6, 5, 2, 6, 1, 18, 3, 8, 9] הפונקציה תחזיר 18  
 לדוגמא עבור הרשימה [6, 5, 2, 6, 1, 6, 3, 8, 9] הפונקציה תחזיר 15  
 דוגמת הדפסה:

Max couple in [6, 5, 2, 6, 1, 3, 8, 9] is 15

## שאלה 2 (35 נקודות):

ממשו את הפונקציה `find_sequence_numbers` המקבלת מטריצה **מלבנית** ומספר שלם חיובי המייצג אורך של רצף. הפונקציה תבדוק את המטריצה, תיצור ותחזיר רשימה דו-ממדית חדשה שכל איבר ברשימה הוא רשימה המכילה את הערך וזוג אינדקסים שמהם מתחיל הרצף של אותו המספר לפי האורך שהתקבל. הבדיקה תבוצע רק **ימינה** (אופקית) ו**למטה** (אנכית).

דוגמאות:

לדוגמא עבור המטריצה  
הבאה ואורך רצף של 4

	0	1	2	3	4
0	1	2	3	4	1
1	0	2	2	2	1
2	1	2	9	0	1
3	1	0	2	4	1
4	4	2	1	1	1

הפונקציה תחזיר את הרשימה  
הבאה:  
[[1, 0, 4], [1, 1, 4]]

לדוגמא עבור המטריצה  
הבאה ואורך רצף של 3

	0	1	2	3	4
0	1	2	3	4	1
1	0	2	2	2	1
2	1	2	9	0	1
3	1	0	2	4	1
4	4	2	1	1	1

הפונקציה תחזיר את הרשימה  
הבאה:  
[[2, 0, 1], [1, 0, 4], [2, 1, 1],  
[1, 1, 4], [1, 2, 4], [1, 4, 2]]

לדוגמא עבור המטריצה  
הבאה ואורך רצף של 2

	0	1	2	3	4
0	1	2	3	4	1
1	0	2	2	2	1
2	1	2	9	0	1
3	1	0	2	4	1
4	4	2	1	1	1

הפונקציה תחזיר את הרשימה  
הבאה:  
[[2, 0, 1], [1, 0, 4], [2, 1, 1],  
[2, 1, 1], [2, 1, 2], [1, 1, 4], [1,  
2, 0], [1, 2, 4], [1, 3, 4], [1, 4,  
2], [1, 4, 3]]

### הנחיות ודגשים לפתרון:

- ניתן לכתוב פונקציות עזר.
- אין לשנות את המטריצה.
- בכל מעבר על תא במטריצה יש לבדוק רק ימינה ולמטה, עבור כל הצלחה יש להוסיף את המיקום, כלומר אם יש רצף ימינה יש להוסיפו לתוצאה ואם קיים מאותו תא רצף נוסף למטה אז גם אותו יש להוסיף לתוצאה, המשמעות שיכול להיות שיהיו לנו רשימות זהות ברשימת התוצאה.

### שאלה 3 (30 נקודות):

נתונים לכם המילון הבא (רק חלק ממנו, המילון המלא קיים בקובץ ב-vpl):

```
songs = {
    "Rocket Man": {
        "writers": ["Elton John", "Bernie Taupin"],
        "performer": ["Elton John"]
    },
    "Someone Like You": {
        "writers": ["Adele"],
        "performer": ["Adele"]
    },
    .....
    "Thinking Out Loud": {
        "writers": ["Ed Sheeran"],
        "performer": ["Ed Sheeran"]
    },
}
```

המפתח של המילון הוא שם של שיר והערך הוא מילון עם 2 מפתחות, המפתח הראשון הוא כותבי השיר והערך הוא רשימה של מי שכתב את השיר, המפתח השני הוא מבצעי השיר והערך הוא רשימה של מבצעי השיר.

### סעיף א (15 נקודות):

ממשו את הפונקציה create\_dict1 המקבלת את המילון של השירים ומחזירה מילון חדש שהמפתח הוא שם האמן והערך הוא רשימה ובה 2 איברים, האיבר הראשון הוא כמות השירים שהאמן כתב והאיבר השני הוא כמות השירים שהאמן ביצע.

לדוגמא עבור המילון של השירים לעיל הפונקציה תחזיר את המילון הבא:

```
{
    'Elton John': [2, 1],
    'Bernie Taupin': [2, 0],
    'Adele': [3, 3],
    'Leonard Cohen': [1, 1],
    'John Lennon': [1, 1],
    'Celine Dion': [0, 1],
    'Pharrell Williams': [1, 1],
    'Freddie Mercury': [1, 1],
    'Luis Fonsi': [1, 1],
    'Erika Ender': [1, 0],
    'Daddy Yankee': [1, 0],
    'Ed Sheeran': [1, 1],
}
```

### סעיף ב (15 נקודות):

ממשו את הפונקציה max\_writer\_and\_performer המקבלת את המילון שנוצר בסעיף א ומחזירה רשימה עם 4 איברים, האיבר הראשון הוא שם האומן שכתב הכי הרבה שירים והאיבר השני הוא כמות

השירים שהוא כתב,  
האיבר השלישי הוא שם האומן שביצע הכי הרבה שירים והאיבר הרביעי הוא כמות  
השירים שהוא ביצע.

לדוגמא עבור המילון לעיל שנוצר בסעיף א הפונקציה תחזיר את הרשימה:

```
['Adele', 3, 'Adele', 3]
```

```
max writer Adele with 3 songs  
max performer Adele with 3 songs
```

### **שאלה 4 (15 נקודות):**

ממש את הפונקציה `is_valid_email` המקבלת כתובת דואר אלקטרוני ומחזירה  
ערך בוליאני מתאים אם כתובת הדואר האלקטרוני מקיימת את התנאים הבאים:

- בדואר האלקטרוני קיים התו `@` פעם אחת בדיוק.
- אורך כתובת הדואר האלקטרוני לפחות 8 תווים ומקסימום 30 תווים.
- שם המשתמש חייב להתחיל באות.
- שם המשתמש מכיל לפחות אות אחת גדולה ולפחות אות אחת קטנה.
- שם השרת מכיל לפחות נקודה אחת.
- שם השרת חייב להסתיים בשתי אותיות לפחות.

דוגמאות:

עבור בדיקת `John.example.com` יוחזר `False`, כי שם המשתמש לא מכיל  
בדיוק `@` פעם אחת בדיוק.

עבור בדיקת `john@example.com` יוחזר `False`, כי שם המשתמש לא מכיל  
אות גדולה.

עבור בדיקת `JSmith@example.org` יוחזר `True`.  
עבור בדיקת `Alice.doe.example.net` יוחזר `False`, כי התו `@` לא קיים.  
עבור בדיקת `Bob@example` יוחזר `False`, כי בשם השרת אין נקודה.  
עבור בדיקת `Jane@invalid.1q` יוחזר `False`, כי שם השרת לא מסתיים בשתי  
אותיות לפחות.  
עבור בדיקת `Mike@mydomain.com` יוחזר `True`.