

### 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

1. שימו לב שרק ההרצה האחרונה היא זאת שנבדקת ורק קוד שנשמר ב-vpl. לכן הקפידו שהתוכן יהיה מגובה ע"י שמירה כל כמה דקות.
2. אסור לכתוב קוד שלא בתוך הפונקציה. במידה ויהיה קוד כזה, ירדו 5 נקודות מהציון הסופי של הבחינה.
3. הקוד צריך לעבוד 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.
4. 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):**

`def remove_string(s, l1, n)`

ממשו את הפונקציה **הרקורסיבית** `remove_string` המקבלת מחרוזת, רשימה של מחרוזות וגודל הרשימה ומחזירה את המחרוזת ללא המחרוזות שנמצאות ברשימה.

לדוגמא עבור המחרוזת `'abcdxyzabc'` והרשימה `['ab', 'yz']` והגודל 2 הפונקציה תחזיר את המחרוזת `'cdxc'`

### **שאלה 2 (35 נקודות):**

**הגדרה:** במטריצה "ריבוע של איברים" הם כל ארבעה איברים העונים על האינדקסים הבאים:  
 $[i][i], [i+1][i], [i+1][i+1], [i][i+1]$

### **סעיף א (15 נקודות):**

ממשו את הפונקציה `has_square_with_sum` המקבלת מטריצה של מספרים ומספר, הפונקציה תחזיר 2 ערכים, הערך הראשון הוא אינדקס השורה והערך השני הוא אינדקס העמודה שבה מתחיל ריבוע של מספרים שסכומם זהה למספר שהתקבל.

מספיק להחזיר את ההתאמה הראשונה גם אם יש עוד התאמות במטריצה.  
במידה ולא נמצא ריבוע של מספרים הפונקציה תחזיר את הערכים -1, -1.  
דוגמאות:

עבור המטריצה הזאת והערך 10 הפונקציה תחזיר את האינדקסים 1,1

```
5 8 3 0 9
2 1 2 7 4
3 5 2 8 2
7 7 6 2 8
square with sum 10 starts at [1][1]
```

עבור המטריצה הזאת והערך 17 הפונקציה תחזיר את האינדקסים -1, -1

```
5 8 3 0 9
2 1 2 7 4
3 5 2 8 2
7 7 6 2 8
square with sum 17 starts at [-1][-1]
```

#### סעיף ב (15 נקודות):

ממשו את הפונקציה `mark_frame_with_zeros` המקבלת מטריצה, אינדקס שורה ואינדקס עמודה של התחלת הריבוע (תוצר של הפונקציה בסעיף א).  
הפונקציה תחליף את כל המספרים מסביב לריבוע באפסים.  
שימו לב **לא לחרוג** מגבולות המטריצה.  
דוגמאות:

```
5 8 3 0 9
2 1 2 7 4
3 5 2 8 2
7 7 6 2 8
Not found square with sum of 17
```

```
5 8 3 0 9
2 1 2 7 4
3 5 2 8 2
7 7 6 2 8
square with sum 10 starts at [1][1]
0 0 0 0 9
0 1 2 0 4
0 5 2 0 2
0 0 0 0 8
```

#### שאלה 3 (30 נקודות):

נתונות לכם 2 הרשימות הבאות:

```
students = ['Kobi Mor', 'Oron Shalom', 'Hen Levi', 'Taly Rozen', 'Alon Lahav']
bank_accounts = [
    {"deposits": [100, 250, 50], "withdrawals": [150, 75, 200]},
    {"deposits": [200, 300, 100, 150], "withdrawals": [250, 150, 50]},
    {"deposits": [500, 200], "withdrawals": [300, 250, 100, 50]},
    {"deposits": [100, 50, 200], "withdrawals": [75, 125, 100]},
    {"deposits": [300, 250, 150, 400], "withdrawals": [200, 100]}
]
```

הרשימה הראשונה היא רשימה של שמות סטודנטים, הרשימה השנייה זאת רשימה של מידע על חשבונות הבנק של הסטודנטים.

**שימו לב,** הרשימות זהות באורכן, אין צורך לבדוק זאת.  
איבר בכל רשימה מתאים לאיבר ברשימה השנייה באותו אינדקס.

#### סעיף א (15 נקודות):

ממשו את הפונקציה create\_dict1 המקבלת את 2 הרשימות ומחזירה מילון חדש ובו המפתח הוא שם הסטודנט והערך הוא מילון ובו 2 מפתחות, סכום ההפקדות וסכום המשיכות של כל סטודנט.

לדוגמא עבור 2 הרשימות לעיל הפונקציה תחזיר את המילון הבא:

```
{
    'Kobi Mor': {'sum_deposits': 400, 'sum_withdrawals': 425},
    'Oron Shalom': {'sum_deposits': 750, 'sum_withdrawals': 450},
    'Hen Levi': {'sum_deposits': 700, 'sum_withdrawals': 700},
    'Taly Rozen': {'sum_deposits': 350, 'sum_withdrawals': 300},
    'Alon Lahav': {'sum_deposits': 1100, 'sum_withdrawals': 300},
}
```

#### סעיף ב (15 נקודות):

ממשו את הפונקציה get\_students\_balance המקבלת את המילון שנוצר בסעיף א ומחזירה מילון חדש ובו המפתח הוא שם הסטודנט והערך הוא מאזן חשבון הבנק של הסטודנט.  
לדוגמא עבור המילון לעיל שנוצר בסעיף א הפונקציה תחזיר את המילון הבא:

```
{
    'Kobi Mor': -25,
    'Oron Shalom': 300,
    'Hen Levi': 0,
    'Taly Rozen': 50,
    'Alon Lahav': 800,
}
```

#### שימו לב:

חובה לממש את הפתרון באמצעות שימוש בפונקציה מסעיף א, מימוש ללא שימוש בפונקציה

מסעיף א יפסל ויקבל ניקוד 0.  
במידה ולא מימשתם בהצלחה את הפונקציה בסעיף א, הניחו את קיומה.

**שאלה 4 (15 נקודות):**

`clear_edges(st, l1)`

ממש את הפונקציה `clear_edges` המקבלת מחרוזת ורשימה של תווים.  
הפונקציה תחזיר מחרוזת **ללא** התווים **בקצוות** המחרוזת עבור התווים המופיעים ברשימה.  
**שימו לב** - פתרון תקין יתקבל ע"י שימוש בפונקציות המחלקה `string`.

דוגמאות:

עבור המחרוזת 'hello world' והרשימה ['h', 'l', 'd', 'o'] הפונקציה תחזיר את המחרוזת הבאה  
'ello wor'

עבור המחרוזת 'Have a good day' והרשימה ['g', 'a', 'y', 'd', ' '], הפונקציה תחזיר את המחרוזת  
הבאה: 'Have a goo'