

**בחינת מפמ"ר במדעי המחשב לכיתה ט'
מבוא לסייבר באמצעות שפת פייתון (Python)
מאי 2019 – אייר תשע"ט**

שם התלמיד/ה: _____ הכיתה: _____

תלמיד יקר,

בשאלון זה שני פרקים:

80 נקודות	פרק ראשון (חובה): ענה על כל השאלות 1-4
20 נקודות	פרק שני (בחירה): ענה על אחת מבין השאלות 5-7
100 נקודות	סך-הכול

קרא בעיון את שאלות המבחן וענה עליהן בתשומת לב.

בשאלות שבהן אתה נדרש לכתוב תשובה, כתוב אותה במקום המיועד לכך.

בשאלות שבהן אתה נדרש לבחור תשובה נכונה אחת מבין כמה אפשרויות, הקף בעיגול את התשובה הנכונה.

בדוק את תשובותיך ותקן אותן לפי הצורך לפני מסירת המבחן.

לשאלון מצורף דף עזר ובו סיכום ההוראות של תכנות פייתון.

חומר עזר מותר בשימוש: כל חומר עזר אישי, למעט מחשבון/מחשב שניתן לתכנות.

משך הבחינה – 120 דקות.

הנחיה למורה: בסוף הבחינה יש לאסוף את הבחינה, כולל דף העזר.

הבחינה הותאמה לגרסת פייתון 2.

ההנחיות בשאלון זה מנוסחות בלשון זכר,
אך מכוונות הן לנבחנות והן לנבחנים

פרק ראשון – חובה (80 נקודות)

ענה על כל השאלות 1-4 (לכל שאלה – 20 נקודות).

שאלה 1

לפניך חמישה קטעי קוד בשפת פייתון.

לכל קוד בטבלה כתוב מהו הפלט המתקבל בעמודה הריקה.

	קוד בשפת פייתון	הפלט המתקבל
1.	<pre>num = 2 num = 10 num = num + 3 print "num = ", num</pre>	
2.	<pre>print "I have", 5-1, "apples."</pre>	
3.	<pre>num = 5 num = int(str(num)+"0") * 2 print num</pre>	
4.	<pre>names = ["Miriam", "Amir", "Sara", "Amir"] print names.index("Amir")</pre>	
5.	<pre>str1 = "I loooove python" print str1[:2:] + str1[2:9:] + " " + str1[10::]</pre>	

שאלה 2

לולאה אינ־סופית היא לולאה שבה תנאי העצירה אינו מתקיים לעולם, ולכן הפקודות שבגוף הלולאה מתבצעות אינ־סוף פעמים. לפניך חמישה קטעי תוכניות המכילים לולאות while. עבור כל קטע תוכנית, קבע אם יש בו לולאה אינ־סופית או לולאה סופית. אם הלולאה סופית, רשום מהו הפלט המתקבל (הקפד על פלט מדויק). אם הלולאה אינ־סופית, סמן X בעמודה המתאימה וכתוב איזה תנאי אינו מתקיים.

מס' הקטע	קטע תוכנית	הלולאה סופית הפלט המתקבל	הלולאה היא אינ־סופית? איזה תנאי אינו מתקיים?
1	<pre>k = 0 while k < 6: print k k = k + 1</pre>		
2	<pre>k = 0 while k != 10: print k k = k + 2</pre>		
3	<pre>k = 4 while k != 20: print k k = k + 5</pre>		
4	<pre>k = 60 while k > 10: print k k = k / 2</pre>		
5	<pre>k = 7 while k == 7: print k k = k % 7</pre>		

שאלה 3

בסיסי ספירה

א. (7 נק')

איך מיוצג המספר העשרוני 374 בבסיס ההקסה־דצימלי? _____ .
הסבר את שלבי הפתרון במילים שלך:

ב. (7 נק')

איך מיוצג המספר ההקסה־דצימלי CD1 בבסיס עשרוני? _____ .
הסבר את שלבי הפתרון במילים שלך:

ג. (6 נק')

1. (1 נק') כתוב מספר בינארי זוגי שאורך ספרותיו גדול מ-4. _____ .
2. (1 נק') כתוב מספר בינארי אי־זוגי שאורך ספרותיו גדול מ-4. _____ .
3. (4 נק') מה משותף לכל המספרים הבינאריים הזוגיים? הקף בעיגול את התשובה הנכונה:
 - א. הספרה המשמעותית ביותר של כל מספר בינארי זוגי היא 1
 - ב. הספרה הפחות משמעותית של כל מספר בינארי זוגי היא 1
 - ג. הספרה המשמעותית ביותר של כל מספר בינארי זוגי היא 0
 - ד. הספרה הפחות משמעותית של כל מספר בינארי זוגי היא 0

הבהרה: במספר 1234 הספרה 4 היא הפחות משמעותית (משום שערכה הוא הנמוך ביותר), והספרה 1 היא המשמעותית ביותר (משום שערכה הוא הגבוה ביותר).

שאלה 4

(5 נק') א. נתון קטע קוד:

```
numbers = []  
for index in range(6):  
    numbers.append((index+1) * 10)  
print numbers
```

מה יודפס בסיום הרצת קטע הקוד? הקפד להציג את איברי הרשימה בהתאם לסדר הדפסתם.

(5 נק') ב. נתון קטע קוד:

```
numbers = [11, 22, 33, 44, 55, 66, 77]  
for index in numbers:  
    if index % 2 == 0:  
        print index
```

מה יודפס בסיום הרצת קטע הקוד? הקפד להציג את איברי הרשימה בהתאם לסדר הדפסתם.

(5 נק') ג. נתון קטע קוד:

```
numbers = [11, 22, 33, 44, 55, 66, 77]  
for index in range(len(numbers)):  
    if index % 2 == 0:  
        print index
```

מה יודפס בסיום הרצת קטע הקוד? הקפד להציג את איברי הרשימה בהתאם לסדר הדפסתם.

(5 נק') ד. נתון קטע קוד:

```
numbers = [10, 11, 23, 31, 42, 55, 60]
for index in range(len(numbers)-1):
    numbers[index+1] = numbers[index]
print numbers
```

מה יודפס בסיום הרצת קטע הקוד? הקפד להציג את איברי הרשימה בהתאם לסדר הדפסתם.

פרק שני – בחירה (20 נקודות)

ענה על שאלה אחת מתוך השאלות 5–7.

שאלה 5

נתונה רשימה המייצגת את סל הקניות של יסמין.

הערכים ברשימה מופיעים בסדר הזה: שם המוצר, מחיר המוצר, שם המוצר, מחיר המוצר וכן הלאה.

לפניך קטע קוד אשר מבצע פעולות על רשימה של מוצרים במבנה שתואר לעיל, ומדפיס את הסכום לתשלום עבור המוצרים, וכן את מספר המוצרים שנרכשו.

לדוגמה:

```
lst = ["Milk", 5.9, "Bread", 10, "Banana", 12.5, "Apples", 15.3]
```

הסכום לתשלום שיודפס : 43.7 ש"ח עבור ארבעה מוצרים.

השלם בקטע הקוד את החלקים החסרים.

שים לב: אורך הרשימה אינו ידוע ולכן מופיע הסימן ... (שלוש נקודות) בסוף הרשימה lst .

```
lst = ["Milk", 5.9, "Bread", 10, "Banana"...]
```

```
summ = _____
```

```
for _____ :
```

```
    summ = _____
```

```
print "הסכום לתשלום הוא", summ
```

```
print "כמות המוצרים שנקנו היא", _____
```

שאלה 6

הוחלט להצפין מסר כך:

כל שני תווים צמודים יוחלפו במקומם.

לדוגמה, עבור המשפט הזה:

The only source of knowledge is experience

המשפט המוצפן יהיה:

hT enoYls uocr efok onlwdeegi sxeePirneec

(10 נק') א. הצפן את המשפט הזה:

Good Luck!

(10 נק') ב.

לפניך קטע קוד חלקי, הכולל פונקצייה אשר מקבלת מחרוזת, ממירה אותה לרשימה, ומדפיסה את הרשימה המוצפנת לפי הצופן הנתון בשאלה. השלם את ההוראות החסרות.

שים לב: הפקודה `list(text)` ממירה את המחרוזת שהתקבלה לרשימה. לדוגמה: עבור: `text = "hello world"` תקבל הרשימה: `[h, e, l, l, o, , w, o, r, l, d]`

```
def exchange(text):  
    text = list(text)  
    for index in range(0, _____, 2):  
        text[index], text[index+1] = text[_____], text[_____]  
    print _____
```


שאלה 7

יעל, מדענית בכירה במכון למחקר ביולוגי בישראל, עובדת בתחום פיתוח החיסונים. יעל ביקשה סיוע בכתיבת קוד המקבל שתי מחרזות המייצגות את קידוד הרצפים הגנטיים של שני חיסונים שונים. בכל רצף גנטי, כל תו מופיע פעם אחת בלבד. המחרזות הראשונה מייצגת את הרצף הגנטי של החיסון לצינון (cold), ואילו המחרזות השנייה מייצגת את הרצף הגנטי של החיסון למחלה בשם ליים (lyme).

יעל רוצה לבדוק אם קיימת קרבה גנטית בין שני החיסונים. קרבה גנטית מוגדרת כך: **בין שני רצפים גנטיים המייצגים חיסונים צריכים להיות לפחות חמישה תווים משותפים** (התווים אינם חייבים להופיע בסדר רציף).

לדוגמה:

```
mkTsAaFlLctfSpQcGJ
```

```
kkLsPalTtFSpnFINc
```

קיימים 12 תווים המשותפים לשתי המחרזות, כלומר, קיימת קרבה גנטית בין

החיסונים.

לפניך קטע קוד שמוצא את כמות התווים המשותפים לשתי המחרזות. השלם את הקוד שלפניך.

```
cold = "mksAaLctTfFSpQGJ"
```

```
lyme = "kKlsPalLctTfFSpnIN"
```

```
count = _____
```

```
for c in cold:
```

```
    found = c in _____
```

```
    if _____:
```

```
        count = _____
```

```
if _____:
```

```
    print "החיסון יעיל"
```

```
else:
```

```
    print "החיסון אינו יעיל"
```

בהצלחה!