

ת"ז:



הchg למערכות מידע ניהול  
תכנות בסיסי (בחינות סטודנט)  
מועד א' סטודנט א' - תשפ"ד  
מרצה: ד"ר רמי רשקוביץ

משך הבחינה: סה"כ 3 שעות

השימוש בכל חומר עזר כתוב מותר

**הנחיות לנבחן:**

- יש לסמן במפורש קטעי טויטה, אחרת אלו ייבדקו וייחשבו כחלק מהפתרונות.  
אין להעביר חומר מסטודנט לסטודנט במהלך הבחינה או לדבר עם סטודנטים אחרים  
שים לב לקיום כל העמודים בטופס הבחינה ולרציפות מספרייהם  
קראו היטב את השאלות והתייחסו לכל הפרטים
- .1
  - .2
  - .3
  - .4

ניקוד כולל	
/20	שאלה 1
/20	שאלה 2
/20	שאלה 3
/20	שאלה 4
/20	שאלה 5
/10	שאלה 6
<b>/110</b>	<b>סיכום</b>

**בהצלחה !**

**הנחיות**

עליכם לפתור את 5 השאלות הבאות. כל שאלה במשקל 20 נקודות. בנוסף, יש שאלת בונוס בשווי 10 נקודות בסוף המבחן. ציון מקסימלי בבחינה - 100

**שאלה 1 (20 נק')**

נתונה הפונקציה הבאה:

```
def mystery(s:str) -> str:
    s = s.lower()
    x = s[0]
    y = s.count(x)
    for c in s:
        if s.count(c) > y:
            x = c
            y = s.count(c)
    return x
```

- המתוודה () lower() מחזירה מחרוזת עם כל התווים באנגלית מסווג lowercase. לדוגמה עבור .'abba'.lower() תחזיר המחרוזת 'Abba No. 1 band'.
- המתוודה count() סופרת את מספר הפעמים בהם מופיע תו/תת-מחרוזת במחרוזת. לדוגמה ('aba'.count('a') תחזיר 2 (הטו 'a' מופיע במחרוזת פעמיים).

- מה יחזיר הפונקציה עבור הקריאה ("She was very successful!") ? mystery('She was very successful!')
- מה עושה הפונקציה? יש לתאר במשפט קצר את המהות ולא את האלגוריתם!  
לדוגמה: הפונקציה מחשבת ומחזירה את מספר התווים באנגלית של המחרוזת שהתקבלה כפרמטר

**פתרונות**

- يُחזיר هنا 's' .
- הפונקציה מוצאת את התו שופיע הכי הרבה פעמים במחרוזת שהתקבלה כפרמטר ומחזירה אותו, בהתאם למבדלי case

**שאלה 2 (20 נק')**

נתונה הפונקציה הבאה:

```
def mystery(a:int, b:int) -> int:
    x = 0
    while a >= b:
        x += 1
        a -= b
    return x
```

- מה יחזיר הפונקציה עבור הפקודה הבאה ? mystery(13,5)
- מה עושה הפונקציה? יש לתאר במשפט קצר את המהות ולא את האלגוריתם!(15 נק')

**לדוגמה: הפונקציה מחשבת ומוחזירה את מספר התווים באנגלית של המחרוזת שהתקבלה**

**כפרמטר**

**פתרונות**

**א. יוזר 2.**

**ב. הפונקציה מחשבת את החלק השלים של חלוקת  $b/a$ , כלומר את  $b//a$**

**שאלה 3 (20 נק')**

כתבו פונקציה complement(`numbers:list`) -> `list` המקבלת כפרמטר רשימה של מספרים שלמים חיוביים בטווח של 1-10 ותחזיר רשימה של כל המספרים בין 1-10 שלא מופיעים ברשימה זו. לדוגמה עבור הרשימה [1,2,4,8] תוחזר הרשימה [3,5,6,7,9,10]. ניתן להניח שהפרמטר תקין.

הערה: מותר לפתרור את השאלה בכל דרך שתמצאו לנכון להשתמש בה.

**פתרונות**

```
def complement(numbers:list) -> list:
    newlist = []
    for num in range(1,11):
        if num not in numbers:
            newlist.append(num)
    return newlist
```

**שאלה 4 (20 נק')**

כתבו תוכנית פיתון אשר קולעת מהמשתמש מספר שלם חיובי כלשהו ולאחר מכן מדפסה את כל המספרים מ-1 ועד המספר שנקלט (כולל) בחלוקת לשורות, קר' שהשורה הראשונה כוללת מספר אחד, השורה השנייה כוללת שני מספרים, וכל שורה נוספת (למעט أول השורה הראשונה) כוללת מספר אחד יותר מהקודמת. ניתן להניח שהקלט מהמשתמש תקין וגודול מ-0.

להלן 2 דוגמאות להרצת התוכנית:

דוגמה 1 :

Enter a number: 6

1  
2 3  
4 5 6

דוגמה 2 :

Enter a number: 17

1  
2 3  
4 5 6  
7 8 9 10  
11 12 13 14 15  
16 17

הערה: שימו לב היטב לדוגמה ולמבנה הדרישה, ובפרט לשורה האחורונה!

**פתרונות**

```

n = int(input('enter a positive integer: '))
num = 1
line = 1
while num <= n:
    for j in range(line):
        print(num, end=' ')
        num += 1
    if num > n:
        break
    print()
    line += 1

```

**שאלה 5 (20 נק')**

כתבו פונקציה `list -> divideBy(numbers:list, divisors:list)` אשר מקבלת **מספרים** 2 רשימות של מספרים שלמים. הפונקציה מחזירה רשימה של כל המספרים ברשימה הראשונה שמתחלקים ללא שארית בכל המספרים ברשימה השנייה. בנוסף, כתבו תוכנית המפעילה את הפונקציה הנ"ל עם 2 רשימות כלשהן (אין צורך בקלט מהמשתמש) ומדפסה את הערך המוחזר ממנה.

להלן שתי דוגמאות להפעלת הפונקציה:

דוגמה 1: עבור הרשימות `[2,3,5,6,8,9,12,14,18,20,40]` ו- `numbers=[2,3,6,12,18]`, `divisors=[2,3]`, תוחזר הרשימה `[6,12,18]` מכיוון שמתוך הרשימה הראשונה הם המספרים היחידים שמתחלקים ללא שארית גם ב-2 וגם ב-3.

דוגמה 2: עבור הרשימות `[2,3,5,6,8,9,12,14,18,20,40]` ו- `numbers=[2,3,4,10]`, `divisors=[2,4,10]`, תוחזר הרשימה `[20,40]` מכיוון שמתוך הרשימה הראשונה הם המספרים היחידים שמתחלקים ללא שארית גם ב-2 וגם ב-4 וגם ב-10.

**פתרונות**

```

def divideBy(numbers:list, divisors:list) -> list:
    newlist = []
    for n in numbers:
        dividedByAll = True
        for d in divisors:
            if n%d != 0:
                dividedByAll = False
                break
        if dividedByAll:
            newlist.append(n)
    return newlist
print(divideBy([1,2,3,5,6,7,9],3))

print(divideBy([2,3,5,6,8,9,12,14,18,20],[2,3]))

```

**שאלה 6 - שאלת בונוס (10 נק')**

כתבו פונקציה `int sumDigits(int n)` אשר מקבלת כפרמטר מספר שלם ומחזירה את סכום ספרותיו. לדוגמה עבור פרמטר `n=124` יוחזר 7 שהוא סכום הספרות 1,2 ו-4. אסור להפוך את המספר למחרוזת, ויש להתייחס אליו כמספר שלם בלבד. בנוסף, כתוב תוכנית שמקבלת כקלט מהמשתמש מספר שלם, ומדפיסה את סכום ספרותיו בעזרת הפונקציה הנ"ל. ניתן להניח שהקלט של המשתמש תקין ומתאים למספר שלם אי-שלילי.

**פתרונות**

```
def sumDigits(n:int) -> int:  
    sum = 0  
    while (n != 0):  
        sum = sum + (n % 10)  
        n = n // 10
```

**בצלחה !**