

ת"ז:



הitag למערכות מידע ניהול
תכנות בסיסי (בחינת סטטוס)
פתרון בחינה לדוגמה

מרצה: ד"ר רמי רשקוביץ

משך הבחינה: סה"כ 3 שעות

השימוש בכל חומר עזר מותר

בהצלחה !

שאלה 1 (20 נק')

כתב פונקציה אשר מקבלת פרמטר מספרי ומדפיסה משולש של מספרים על פי התבנית. למשל אם המספר שהועבר כפרמטר הוא 7, הפונקציה תדפיס:

```

1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
6 12 18 24 30 36
5 10 15 20 25
4 8 12 16
3 6 9
2 4
1

```

```

def q1(n):
    if n < 1: return
    for i in range(1,n+1):
        for j in range (1,i+1):
            print(i*j, ' ', end="")
        print()
    for i in range(n-1,0,-1):
        for j in range(1,i + 1):
            print(i*j, ' ', end="")
        print()

```

שאלה 2 (20 נק')

כתב פונקציה אשר מקבלת פרמטר שהוא מחרוזת המייצגת משפט. הפונקציה תחזיר True אם המחרוזת היא פנגרמה – False אם לא. פנגרמה היא משפט הכלול את כל אותיות האלפבית (כל אות לפחות פעם אחת – בהתעלם מגודל האות). למשל, כל אחד מהמשפטים הרצ"ב הוא פנגרמה:

- The quick brown fox jumps over the lazy dog
- The five boxing wizards jump quickly
- Jackdaws love my big sphinx of quartz

```

def q2(word):
    alphabet ='abcdefghijklmnopqrstuvwxyz'
    word = word.lower()
    for ch in alphabet:
        if ch not in word:
            return False
    return True

```

שאלה 3 (20 נק')

ההפרש הסימטרי של 2 קבוצות A ו-B מוגדר כאיברים שהייכים לקבוצה A ולא שייכים לקבוצה B, ואיברים השייכים לקבוצה B שלא שייכים לקבוצה A, כמובן, כל האיברים שלא נמצאים בחיתוך של A ו-B. כתוב פונקציה בשם `symetricDifference` המקבלת כפרמטרים שתי רשימות A ו-B המיצגות 2 קבוצות של מספרים שלמים ומחזירה רשימה שלישית המכילה את ההפרש הסימטרי של A ו-B. אין חשיבות לסדר המספרים בתוצאה.

שים לב: קבוצה מורכבת מאיברים שונים ללא חזרות. נתון ש-A ו-B הן קבוצות!

דוגמת הרצות:

```
symetricDifference([3, 4, 5, 6, 7], [2, 4, 6, 8])      >>> [2, 3, 5, 7, 8]
symetricDifference([1, 2, 3], [6, 7])                  >>> [1, 2, 3, 6, 7]
symetricDifference([1, 2, 3], [1, 2, 3])                >>> []
symetricDifference([], [1, 2, 3])                      >>> [1, 2, 3]
symetricDifference([1, 2, 3], [1, 2])                  >>> [3]
```

```
def SymetricDiff(A, B):
    result=[]
    for a in A:
        if a not in B:
            result.append(a)

    for b in B:
        if b not in A:
            result.append(b)

    return result
```

שאלה 4 (20 נק')

כתב פונקציה בשם `FractionalDecrease` אשר מקבלת שני פרמטרים שהם מספרים שלמים מייצגים מונה ומכנה של שבר. הפונקציה תחזיר את המונה והמכנה של השבר לאחר שהוא צומצם, או אם אין אפשרויות לצמצמו תחזיר את המונה והמכנה המקוריים. שים לב: הפונקציה מחזירה שני ערכים המייצגים את המונה והמכנה.

דוגמת הרצות:

```
FractionalDecrease(123, 492)      >>> 1, 4
FractionalDecrease(1, 5)           >>> 1, 5
FractionalDecrease(999, 111)       >>> 9, 1
FractionalDecrease(9, 15)          >>> 3, 5
FractionalDecrease(99, 33)         >>> 3, 1
FractionalDecrease(24, 60)         >>> 2, 5
FractionalDecrease(32, 128)        >>> 1, 4
```

```

def FractionalDecrease(nominator, denominator):
    minimum = min(nominator,denominator)
    for i in range (minimum, 0, -1):
        if (nominator % i) == 0 and (denominator % i) == 0:
            return nominator // i, denominator // i

print(FractionalDecrease(123,492) )
print(FractionalDecrease(99,33))

```

 שאלה 5 (20 נק')

כתבו תכנית המתקבלת מהשימוש שני מספרים טבעיים n - m , מחשבת ומדפיסה את כל המספרים הראשוניים בטוח מ- n עד m ואת סכומם.
מספר ראשוני הוא מספר טבעי המתחלק ללא שארית רק בעצמו ובאחד.
שים לב: ניתן להניח שהקלט תקין ומתקיים $n < m$.
דוגמת הרצות:

```

Enter first number: 2
Enter second number: 9
Prime numbers between 2 and 9: [2,3,5,7] (sum is 17)

```

```

Enter first number: 10
Enter second number: 12
Prime numbers between 10 and 12: [11] (sum is 11)

```

```

Enter first number: 10
Enter second number: 20
Prime numbers between 10 and 20: [11,13,17,19] (sum is 60)

```

```

def isPrime(n):
    if n == 1:
        return False
    elif n==2:
        return True

    for i in range (2,n):
        if (n % i) == 0:
            return False
    return True

def primes(n,m):
    sum = 0
    prime_numbers = []
    for i in range (n,m+1,1):
        if isPrime(i):
            prime_numbers.append(i)
            sum += i
    print (prime_numbers, ' (sum is', sum, ')')

primes(1,50)

```