

**פתרון למבחן בקורס מבוא מורחב למדעי המחשב, CS1001.py**

**סמסטר ב' 2013**

**מועד ב**

**שאלה 1 (20 נק')**

א. (14 נק')

1. ['a', 'b', 'c', 'd', (3,5), 'x', (6,5), 'x']

לא ייתכן, משום שתמיד מחפשים את ההתאמה הארוכה ביותר, ולכן היה צריך להיות

['a', 'b', 'c', 'd', (3,5), 'x', (6,6)]

טעות נפוצה היתה לטעון שחזרה אחת לא יכולה לגלוש לתוך תווים שנדחסו בחזרה מוקדמת יותר.

2. ['a', 'b', 'c', 'd', (3,20)]

המחרוזות לפני הדחיסה: 'a(bcd)<sup>7</sup>bc'

ב. (6 נק') אפשרויות להשלמה: abcccc, abcbebc, abcabca

**שאלה 2 (20 נק')**

א. (4 נק')  $2^3 = 8$

ב. (4 נק')  $d=4$

ג. (12 נק')

הודעה מקורית (אם אפשר לשחזר)	מספר השגיאות	שדר שנתקבל
<u>אי אפשר לשחזר</u>	<u>2</u>	1 1 1 0 1 1 1 0
<u>010</u>	<u>1</u>	0 1 0 1 1 1 0 1
<u>001</u>	<u>0</u>	0 0 1 1 0 0 1 1

**שאלה 3 (20 נק')**

א. (10 נק')

```
def find_secret(N1, N2):  
    low = N1  
    high = N2  
    ans = "?"  
    while ans != "=":  
        mid = (low + high) // 2  
        ans = guess(mid)  
        if ans == ">":  
            low = mid + 1  
        else:  
            high = mid - 1  
    return mid
```

ב. (10 נק')

```
def find_secret2():  
    low = 1  
    high = 2  
    answer = guess(high)  
    while answer == ">":  
        low *= 2  
        high *= 2  
        answer = guess(high)  
  
    if answer == "=":  
        return high  
    return find_secret(low, high)
```

## שאלה 4 (20 נק')

שאלה זאת עוסקת במספרים שלמים, הגורמים הראשוניים שלהם, והמחלקים שלהם.

א. (7 נק')

```
def factors(n,start=2):
    # assert - n does not have a factor that is smaller than start

    while n%start !=0 and n > start : # התנאי השני נחוץ כתנאי סיום
        start +=1

    if start>n:
        return []
    else:
        return [start] + factors(n//start, start) # אפשר להשמיט את הפרמטר השני בקריאה
```

ב. (6 נק')

```
def areAmicable(m, n):
    return m!=n and m == sumDivisors(n) and n == sumDivisors(m)
    לפי ההגדרה המספרים חייבים להיות שונים
```

ג. (7 נק')

```
def allAmicable():
    n = 2 # 220 ok too

    while True:
        m = sumDivisors(n)

        if sumDivisors(m) == n and m>n: # כדי שכל זוג יופיע פעם אחת
            yield n,m

        # increment

        n +=1
```

## שאלה 5 (20 נקודות)

א. (10 נק')

```
import math
class Circle:
    # Constructor:
    def __init__(self, R, a, b):

        self.R = R

        self.a = a

        self.b = b

    def __repr__(self):
        return "Circle center " + self.a + ", " + self.b + " with radius " + self.radius

    def area(self):

        return math.pi * (self.R ** 2)

    def circumference(self):

        return 2 * math.pi * self.R

    def __mul__(self, factor):

        assert type(factor) in (int, float)

        return Circle(self.R * factor, self.a, self.b)

    def move(self, point):

        self.a = point[0]

        self.b = point[1]

    def contains (self, point):

        dist_sq = (self.a - point[0]) ** 2 + (self.b - point[1]) ** 2

        return self.R ** 2 >= dist_sq

    def intersect(self, other):

        dist_centers_sq = (self.a - other.a) ** 2 + (self.b - other.b) ** 2

        return dist_centers_sq < (self.R + other.R) ** 2
```

ב. (4 נק') רציים רק על כל הפיקסלים שבתוך ריבוע בעל גודל צלע  $2R$ , סה"כ  $4R^2$  פיקסלים שלכל אחד מהם מבצעים  $O(1)$  פעולות. לכן  $O(R^2)$ .

ג. (6 נק')

```
def paint (self, img):
    n,m = img.dim()
    center = (n // 2, m // 2)
    for i in range(center[0] - self.R, center[0] + self.R + 1):
        for j in range(center[1] - self.R, center[1] + self.R + 1):
            if (center[0] - i) ** 2 + (center[1] - j) ** 2 <= self.R ** 2:
                img[i, j] = 0
```

**סוף!**