

Started on	Friday, 1 January 2021, 7:58 PM
State	Finished
Completed on	Friday, 1 January 2021, 10:54 PM
Time taken	2 hours 55 mins
Marks	6.40/7.00
Grade	9.14 out of 10.00 (91%)



Question 1

Correct

Mark 1.00 out of 1.00

İlkköşegen toplamı: Matrisin ilkköşegendeki elemanları toplamıdır.

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

Yukarıda verilen matris üzerinde ilkköşegen 1-6-11-16 doğrultusundadır ve ilkköşegen toplamı $1 + 6 + 11 + 16 = 34$ olarak bulunur

Sizden istenilen iç içe listeler halinde verilen girdi için **trace** isimli bir fonksiyon yazmanız ve ekrana ilkköşegen toplamını yazdırmanız.

Örnek: Yukarıda verilen matris, iç içe listeler kullanarak `[[1, 15, 14, 4], [12, 6, 7, 9], [8, 10, 11, 5], [13, 3, 2, 16]]` şeklinde gösterilir ve ilkköşegen toplamı olarak ekrana 34 yazdırılmalıdır.

Answer: (penalty regime: 10, 20, ... %)

```
1 def trace(liste):
2     sums = 0
3     for i in range(len(liste)):
4         sums += liste[i][i]
5     print(sums)
```

	Test	Expected	Got	
✓	trace([[1, 15, 14, 4], [12, 6, 7, 9], [8, 10, 11, 5], [13, 3, 2, 16]])	34	34	✓
✓	trace([[1, 15, 14], [12, 6, 7], [8, 10, 11]])	18	18	✓
✓	trace([[1, 15], [12, 6]])	7	7	✓
✓	trace([[1]])	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 0.50 out of 1.00

Sihirli kare: iç içe listeler halinde verilen, 2 boyutlu girdinizin her sırasının, her kolonunun ve her iki köşegeninin toplamaları birbirine eşit ise bu kareye **sihirli kare** diyoruz.

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

Yukarıda verilen matris üzerinde satır, sütun ve iki köşegen toplamalarına bakalım:

1. Satır : $1 + 15 + 14 + 4 = 34$

2. Satır : $12 + 6 + 7 + 9 = 34$

3. Satır : $8 + 10 + 11 + 5 = 34$

4. Satır : $13 + 3 + 2 + 16 = 34$

1. Kolon: $1 + 12 + 8 + 13 = 34$

2. Kolon: $15 + 6 + 10 + 3 = 34$

3. Kolon: $14 + 7 + 11 + 2 = 34$

4. Kolon: $4 + 9 + 5 + 16 = 34$

1. Köşegen (sol üstten, sağ alta) : $1 + 6 + 11 + 16 = 34$

2. Köşegen (sol alttan, sağ üste) : $13 + 10 + 7 + 4 = 34$

Gördüğünüz üzere bütün toplamalar birbirine eşit olduğu için matrisimiz bir sihirli karedir.

Yazacağınız **isSihirli** fonksiyonu, girdi olarak iç içe listeler halinde tanımlanmış bir liste almalıdır, eğer girdi sihirli kare ise ekrana 0, değil ise 1 bastırmalıdır.

Örnek: `isSihirli([[1, 15, 14, 4], [12, 6, 7, 9], [8, 10, 11, 5], [13, 3, 2, 16]])` ekrana 0 yazdırmalıdır.

Answer: (penalty regime: 10, 20, ... %)

```
1 def isSihirli(liste):
2     rowSums = [0 for i in range(len(liste))]
3     colSums = [0 for i in range(len(liste))]
```

```

4     diagonSums = [0] * 2
5     isMagic = 0
6
7     for i in range(len(liste)):
8         for j in range(len(liste)):
9             rowSums[i] += liste[i][j]
10            colSums[j] += liste[j][i]
11
12    for i in range(len(liste)):
13        diagonSums[0] += liste[i][i]
14
15    for i in range(len(liste)):

```

	Test	Expected	Got	
✓	isSihirli([[8, 1, 6], [3, 5, 7], [4, 9, 2]])	0	0	✓
✓	isSihirli([[8, 1, 7], [3, 5, 7], [4, 9, 2]])	1	1	✓
✓	isSihirli([[1, 15, 14, 4], [12, 6, 7, 9], [8, 10, 11, 5], [13, 3, 2, 16]])	0	0	✓
✓	isSihirli([[1, 22, 14, 4], [12, 6, 7, 9], [8, 10, 11, 5], [13, 3, 2, 16]])	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.50/1.00**.

Question 3

Correct

Mark 1.00 out of 1.00

Bu soruda sizden istenilen girdi olarak 2 tane liste alan ve sonuç olarak ekrana liste yazan **maksList** fonksiyonunu yazmanız. maksList fonksiyonu almış olduğu iki listenin elemanlarını indekslerini artırarak kontrol edip, maksimum elemanlardan oluşan yeni listeyi döndürecektir.

Girdi olarak [4, 5, 6] ve [1, 2, 6] listelerini maksList fonksiyonuna gönderdiğimizde fonksiyonumuz sırasıyla 4 ile 1 elemanlarını, 5 ile 2 elemanlarını ve 6 ile 6 elemanlarını karşılaştırıp maksimum elemanlardan oluşan [4, 5, 6] listesini ekrana bastıracaktır.

Answer: (penalty regime: 10, 20, ... %)

```
1 def maksList(list1, list2):
2     result = []
3
4     for i in range(len(list1)):
5         if list1[i] >= list2[i]:
6             result.append(list1[i])
7         else:
8             result.append(list2[i])
9     print(result)
```

	Test	Expected	Got	
✓	maksList([4, 5, 6], [1, 2, 6])	[4, 5, 6]	[4, 5, 6]	✓
✓	maksList([4, 5], [4, 2])	[4, 5]	[4, 5]	✓

	Test	Expected	Got	
✓	<code>maksList([4], [4])</code>	<code>[4]</code>	<code>[4]</code>	✓
✓	<code>maksList([4, 5, 6, 3], [1, 2, 8, 6])</code>	<code>[4, 5, 8, 6]</code>	<code>[4, 5, 8, 6]</code>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 0.90 out of 1.00

Bu soruda yığıt veri yapısını kullanarak size verilen karakter dizisinin palindrom [bknz] olup olmadığını kontrol etmenizi istiyoruz. Yığıt yapısının çalışma mantığını sizinle paylaşmıştık, eğer Python listelerini kullanmak isterseniz, yığta eleman eklemeyi, ve yığıttan eleman çıkarmayı sıfıncı indeksten yapmalısınız.

Örnek olarak [1, 2, 3, 4, 5] bizim yığıt yapımız için kullandığımız liste ise eleman çıkardığımız zaman [2, 3, 4, 5] listesini, orijinal listeye 6 elemanını eklediğimiz zaman da [6, 1, 2, 3, 4, 5] listesini elde etmemiz gerekiyor.

Yazmanızı istediğimiz fonksiyonun adı isPal olmalıdır, tek bir girdi (karakter dizisi) almalı ve eğer girdi palindrom ise ekrana 0, değil ise 1 yazdırmalıdır.

Answer: (penalty regime: 10, 20, ... %)

```
2 stack = []
3 isPalin = 1
4 inStr = inStr.casefold()
5
6 for ch in inStr:
7     stack.append(ch)
8
9 reverseStr = ''
10 while len(stack) > 0:
11     reverseStr = reverseStr + stack.pop()
12
13 if inStr == reverseStr:
14     isPalin = 0
15
16 print(isPalin)
```

	Test	Expected	Got	
--	------	----------	-----	--

	Test	Expected	Got	
✓	isPal("abba")	0	0	✓
✓	isPal("abca")	1	1	✓
✓	isPal("acaba")	1	1	✓
✓	isPal("ababa")	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.90/1.00**.

Question 5

Correct

Mark 1.00 out of 1.00

Mükemmel Sayı: Eğer pozitif bir tamsayı, kendisi dışında bütün pozitif bölenlerinin toplamına eşit ise o sayıya pozitif tamsayı denilir.

Örnek olarak 28 tamsayısı mükemmel sayıdır, çünkü kendisi dışındaki bölenleri (1, 2, 4, 7, 14) toplamı 28'e eşittir.

Bu soruda sizden istenilen girdi olarak bir tamsayı alan **isPer** fonksiyonunu yazmanızdır. Eğer girdi mükemmel sayı ise ekrana 0, değil ise 1 yazdırmalısınız. Bunu yaparken kuyruk veri yapısını kullanmanızı istiyoruz. Girdi olarak aldığınız sayının kendisi dışındaki pozitif bölenlerini kuyruk veri yapısının kurallarına uygun bir şekilde Python listesine atıp, en sonunda listenin elemanlarının toplamının girdi olarak aldığınız sayıya eşit olup olmadığını kontrol etmelisiniz:

28 sayısı için kuyruk veri yapısına uyan listeniz sırasıyla [1, 2, 4, 7, 14] elemanlarını içermelidir.

Answer: (penalty regime: 10, 20, ... %)

```
1 def isPer(num):
2     queue = []
3     sums = 0
4     isper = 1
5
6     for n in range(1,1+num//2):
7         if num % n == 0:
8             queue.append(n)
9             sums += n
10    if sums == num:
11        isper = 0
12    print(isper)
13
```

Test	Expected	Got	
------	----------	-----	--

	Test	Expected	Got	
✓	isPer(28)	0	0	✓
✓	isPer(30)	1	1	✓
✓	isPer(496)	0	0	✓
✓	isPer(498)	1	1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6

Correct

Mark 1.00 out of 1.00

Bu sorumuzda dilimleme işlemi yapacağız. Girdileriniz sırasıyla iç içe tanımlanmış 2 boyutlu bir liste, dilimlemenin sol indeksi ve dilimlemenin sağ indeksi. Sizden istenilen sol indeksten, sağ indekse kadar olan (sağ indeks dahil değil) listelerin karşılıklı gelen elemanlarının toplamını ekrana liste olarak yazdırmak.

Örnek :

girdi: [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], 1, 3

çıktı: [11, 13, 15]

açıklama: [4, 5, 6] indeksi 1 olan elemandır, [10, 11, 12] indeksi 3 olan elemandır ve dahil değildir. Aralarında kalan [4, 5, 6] ve [7, 8, 9] listelerinin elemanlarını aynı indeksteki elemanlar toplanarak yeni bir liste oluşturup, ekrana yazdırdığımızda [11, 13, 15] listesi görünür.

Fonksiyon adınız **listeDil** olmalıdır, ve üç tane girdi alacak şekilde tanımlamalısınız, işlemlerinizi bittiğinde ekrana elde ettiğiniz listeyi yazdırmalısınız.

Answer: (penalty regime: 10, 20, ... %)

```
1 def listeDil(list1, left, right):
2     result =[0] * len(list1[left])
3
4     for i in range(len(list1[left])):
5         for j in range(left, right):
6             result[i] += list1[j][i]
7     print(result)
```

	Test	Expected	Got	
✓	listeDil([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], 1, 3)	[11, 13, 15]	[11, 13, 15]	✓
✓	listeDil([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12]], 0, 3)	[9, 12]	[9, 12]	✓
✓	listeDil([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]], 0, 1)	[1, 2, 3, 4]	[1, 2, 3, 4]	✓
✓	listeDil([[1], [2], [3], [4], [5], [6], [7], [8], [9]], 0, 7)	[28]	[28]	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7

Correct

Mark 1.00 out of 1.00

Bu soruda **stackOp** isminde bir fonksiyon yazacaksınız, fonksiyonunuz girdi olarak bir liste, operasyon karakteri ve bir tamsayı alacak ve aşağıdaki işlemleri yapacaktır:

Eğer operasyon '+' ise, aldığı tamsayıyı listeye yığıt yapısını bozmadan en baştan ekleyecek ve yeni listeyi ekrana bastıracaktır.

Operasyon '-' ise, yığıtın tepesindeki elemanı yığıttan çıkaracak ve kalan listeyi ekrana bastıracaktır.

Operasyon '?' ise, yığıtın tepesindeki elemanı ekrana bastıracaktır.

stackOp([1, 2, 3, 4, 5], '+', 11) çıktı olarak [11, 1, 2, 3, 4, 5] listesini ekrana yazdıracaktır.

stackOp([1, 2, 3, 4, 5], '-', 10) çıktı olarak [2, 3, 4, 5] listesini ekrana yazdıracaktır. (10 sayısının bir önemi yoktur)

stackOp([1, 2, 3, 4, 5], '?', 10) çıktı olarak 1 tamsayısını ekrana yazdıracaktır. (10 sayısının bir önemi yoktur)

Answer: (penalty regime: 10, 20, ... %)

```
1 def stackOp(list1,opCh, num):
2     if opCh == '+':
3         list1.insert(0, num)
4         print(list1)
5     elif opCh == '-':
6         del(list1[0])
7         print(list1)
8     else:
9         print(list1[0])
```

Test	Expected	Got	
------	----------	-----	--

	Test	Expected	Got	
✓	stackOp([1, 2, 3, 4, 5], '+', 11)	[11, 1, 2, 3, 4, 5]	[11, 1, 2, 3, 4, 5]	✓
✓	stackOp([1, 2, 3, 4, 5], '-', 10)	[2, 3, 4, 5]	[2, 3, 4, 5]	✓
✓	stackOp([1, 2, 3, 4, 5], '?', 10)	1	1	✓
✓	stackOp([5, 4, 3, 2, 1], '?', 10)	5	5	✓
✓	stackOp([5, 4, 3, 2, 1], '+', 11)	[11, 5, 4, 3, 2, 1]	[11, 5, 4, 3, 2, 1]	✓
✓	stackOp([5, 4, 3, 2, 1], '-', 11)	[4, 3, 2, 1]	[4, 3, 2, 1]	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.



PREVIOUS ACTIVITY
Alıştırma - Veri Yapıları

NEXT ACTIVITY

8. Hafta - Python'da Özyinelemeli Yapılar ve Dosya İşlemleri

