



KREDİ KARTI NUMARASI DOĞRULAMA PROGRAMI

- Günlük hayatımızın bir parçası haline gelen **Kredi kartı**, bankaların müşterilerine belirli bir dönemde kullanılmak üzere belli bir miktarda limitle verdiği ve anlaşmalı üye iş yerlerinde POS cihazı üzerinden belirlenen limit ve dönem dahilinde alışverişlerde kullanılabilen bir çeşit **ödeme ve borçlanma aracıdır**.
- Yaygınlaşan kart kullanımına karşı kredi kartı sahteciliği de arttı. Kredi kartı alırken çok dikkatli olmamız gerekmektedir.
- Ben de kartımızın sahte olup olmadığını anlamamız için python dilini kullanarak kart numarası kontrol programı yaptım.
- Yaptığım programda 2 tip doğrulama sistemi kullandım.

1.Tip doğrulama sistemimizin kuralı şu şekildedir:

- En soldaki check digit değerinden başlayarak sola doğru 1., 3., 5.,... değerleri 2 ile çarpılır.
- Eğer değerlerin toplamı 9'dan büyükse rakamlarını toplayıp rakamların toplamını buluruz.
- Nihai olarak elde edilen tüm rakamlar toplanır.
- Eğer tüm toplam 10'un katı ise ilgili kart numarası geçerlidir. 10'un katı değilse geçersiz bir kart numarasıdır.

1.Tip doğrulama sistemimize göre:

Örneğin;

	4	+	2	+	6	+	7	+	5	+	7	+	8	+	6	= 45
5	4	9	2	9	6	8	7	9	5	8	7	4	8	7	6	
5*2=10	9*2=18	9*2=18	8*2=16	9*2=18	8*2=16	4*2=8	7*2=14									
1+0=1	1+8=9	1+8=9	1+6=7	1+8=9	1+6=7	8	1+4=5									
1	+	9	+	9	+	7	+	9	+	7	+	8	+	5	= 55	

45+55=100 (mod10)=0 olduğunda kart numarası geçerli olur.

1.Programımızın kodları ve debug işlemimiz:

```
07luhn_algoritmasi.py
n = 16      n: 16
d1 = {}     d1: {1: 1, 2: 4, 3: 9, 4: 2, 5: 9, 6: 6, 7: 7, 8: 7, 9: 9, 10: 5, 11: 7, 12: 7, 13: 8, 14: 8, 15: 5, 16
d2 = {}     d2: {1: 5, 2: 4, 3: 9, 4: 2, 5: 9, 6: 6, 7: 8, 8: 7, 9: 9, 10: 5, 11: 8, 12: 7, 13: 4, 14: 8, 15: 7, 16
for i in range(1, n + 1):      i: 16
    d2[i] = int(s[i - 1])
for i in range(1, n + 1):
    if i % 2 != 0:
        d1[i] = 2 * d2[i]
        if d1[i] > 9:
            d1[i] = int(d1[i] / 10) + d1[i] % 10
    else:
        d1[i] = d2[i]

tekler_toplami = 0      tekler_toplami: 55
ciftler_toplami = 0     ciftler_toplami: 45
for i in range(1, n + 1):
    if i % 2 != 0:
        tekler_toplami += d1[i]
    else:
        ciftler_toplami += d1[i]

toplam = tekler_toplami + ciftler_toplami

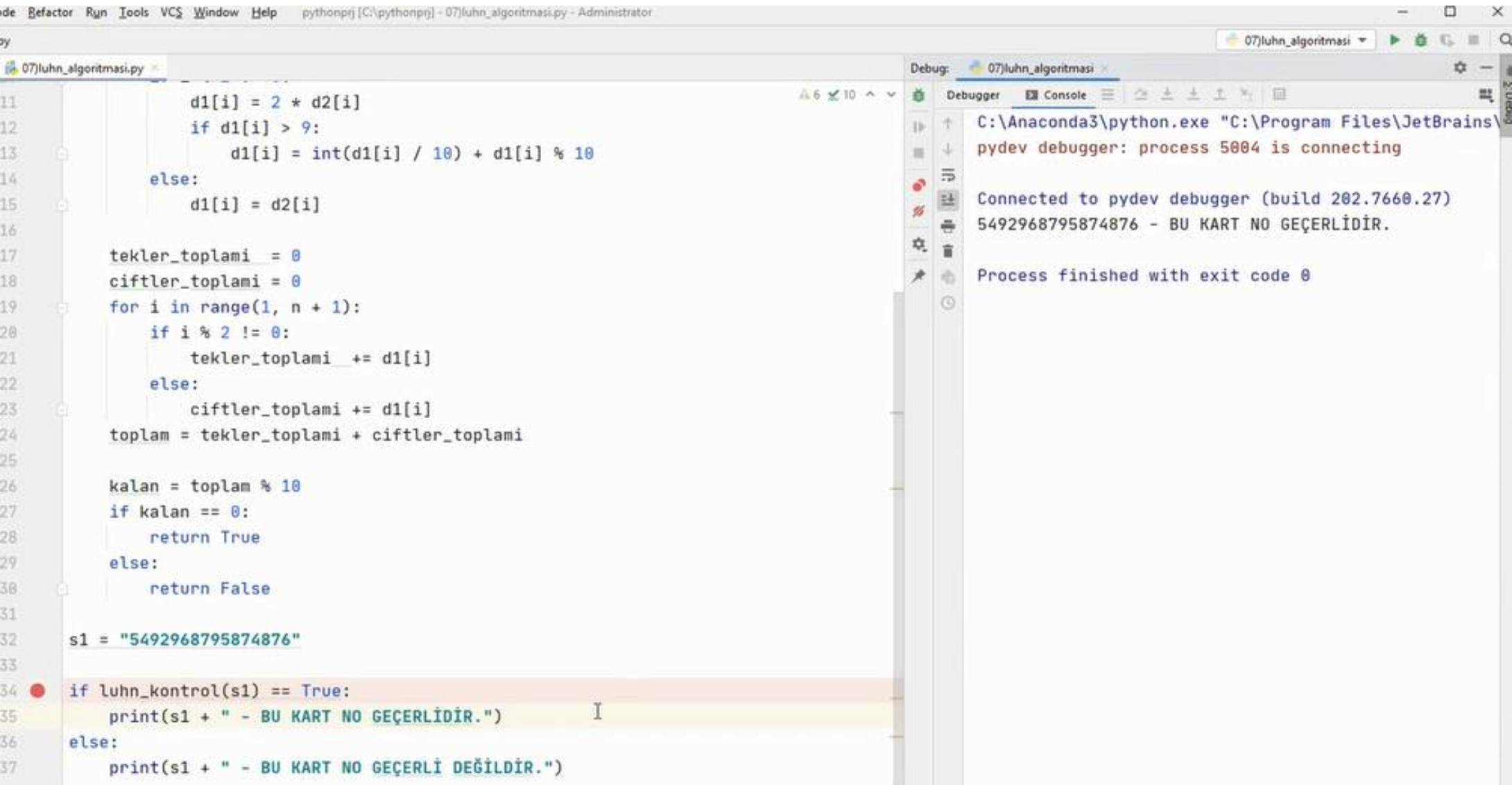
kalan = toplam % 10
if kalan == 0:
    return True
else:
    return False

s1 = "5492968795874876"

if luhn_kontrol(s1) == True:
    print(s1 + " - BU KART NO GEÇERLİDİR.")

luhn_kontrol()
Terminal Python Console
```

Örneğimizin geçerli olup olmadığını kontrol etmek için program ekran çıktımız:



The screenshot shows a Python IDE with a file named '07)luhn_algoritmasi.py'. The code implements the Luhn algorithm to check if a card number is valid. The card number '5492968795874876' is stored in the variable 's1'. The program checks the validity of the card number and prints the result. The output in the console shows that the card number is not valid.

```
11     d1[i] = 2 * d2[i]
12     if d1[i] > 9:
13         d1[i] = int(d1[i] / 10) + d1[i] % 10
14     else:
15         d1[i] = d2[i]
16
17     tekler_toplami = 0
18     ciftler_toplami = 0
19     for i in range(1, n + 1):
20         if i % 2 != 0:
21             tekler_toplami += d1[i]
22         else:
23             ciftler_toplami += d1[i]
24     toplam = tekler_toplami + ciftler_toplami
25
26     kalan = toplam % 10
27     if kalan == 0:
28         return True
29     else:
30         return False
31
32     s1 = "5492968795874876"
33
34     if luhn_kontrol(s1) == True:
35         print(s1 + " - BU KART NO GEÇERLİDİR.")
36     else:
37         print(s1 + " - BU KART NO GEÇERLİ DEĞİLDİR.")
```

Debugger Console Output:

```
C:\Anaconda3\python.exe "C:\Program Files\JetBrains\pydev debugger: process 5004 is connecting
Connected to pydev debugger (build 202.7660.27)
5492968795874876 - BU KART NO GEÇERLİDİR.
Process finished with exit code 0
```

2.Tip doğrulama sistemimizin kuralı şu şekildedir:

- Son digit(son 16.hane) 'i ayırıyoruz. Geriye 15 tane basamağımız kalıyor.
- Bu 15 tane basamağımızı tersten (sağdan sola doğru) yazıyoruz.
- 1.tip doğrulamadaki kuralımızı uyguluyoruz.
- Nihai olarak elimize çıkan toplamın 9 ile çarpımının 10'a bölümünden kalan, ayırdığımız 16. haneyi veriyorsa kart numarası geçerlidir. 16.haneyi vermiyorsa kart numarası geçerli değildir.

Örneğin;

	8	+	7	+	5	+	7	+	6	+	2	+	4	= 39
7	8	4	7	8	5	9	7	8	6	9	2	9	4	5
7*2=14		4*2=8		8*2=16		9*2=18		8*2=16		9*2=18		9*2=18		5*2=10
1+4=5		8		1+6=7		1+8=9		1+6=7		1+8=9		1+8=9		1+0=1
5	+	8	+	7	+	9	+	7	+	9	+	9	+	1 = 55

39+55=94, 94*9=846 (mod 10)=6 son hanemiz de 6 olduğu için kart numarası geçerli olur.

2. Programımızın kodları:

```
def son_digit_dogrulama(s):  
    n = 16  
    d1 = {}  
    d2 = {}  
    for i in range(1, n+1):  
        d2[i] = int(s[i-1])  
    for i in range(1, n):  
        d1[i] = d2[n-i]  
    for i in range(1, n):  
        if i % 2 != 0:  
            d1[i] = 2 * d1[i]  
            if d1[i] > 9: d1[i] -= 9  
    toplam = 0  
    for i in range(1, n):  
        toplam += d1[i]  
    kalan = 9 * toplam % 10  
    if kalan == d2[n]:  
        return True, kalan  
    else:  
        return False, -1  
  
s1 = "5492968795874876"  
sonuc, sondigit = son_digit_dogrulama(s1)  
if sonuc == True:  
    print(s1+ "-BU KART NO GEÇERLİDİR. sondigit: ", sondigit)  
else:  
    print(s1+ "-BU KART NO GEÇERLİ DEĞİLDİR. ")
```

Örneğimizin geçerli olup olmadığını kontrol etmek için program ekran çıktımız:

```
    else:
        return False, -1

s1 = "5492968795874876"
sonuc, sondigit = son_digit_dogrulama(s1)
if sonuc == True:
    print(s1+ "-BU KART NO GEÇERLİDİR. sondigit: ", sondigit)
else:
    print(s1+ "-BU KART NO GEÇERLİ DEĞİLDİR. ")
```

```
5492968795874876-BU KART NO GEÇERLİDİR. sondigit: 6
```