

```

def myfunc(): #Definition
    print("Hello Python Lovers")
myfunc() #Call

Hello Python Lovers

def details(name, userid, country):
    print('Name :- ', name)
    print('User ID is :- ', userid)
    print('Country :- ', country)

details('Kuzey', 'kuzey12345', 'Norway')

Name :- Kuzey
User ID is :- kuzey12345
Country :- Norway

def square(n):
    n = n * n
    return n

square(10)

100

#Docstring
def square(num):
    '''Square Function : This function will return the square of a
number'''
    return num**2

square(2)

4

square.__doc__

'Square Function : This function will return the square of a number'

def even_odd(num):
    """This function will check whether a number is even or odd"""
    if num %2 == 0:
        print(num, "is an even number")
    else:
        print(num, "is an odd number")

even_odd(3)
even_odd(4)
print(even_odd.__doc__)

```

3 is an odd number
4 is an even number
This function will check whether a number is even or odd

#String Birleştirme

```
def fullname(firstname, middlename, lastname):  
    fullname = "{} {} {}".format(firstname, middlename, lastname)  
    print(fullname)
```

```
fullname('Kuzey', 'Batı', 'Güney') #positional argument
```

Kuzey Batı Güney

```
def fullname(firstname, middlename, lastname):  
    fullname = "{} {} {}".format(firstname, middlename, lastname)  
    print(fullname)
```

```
fullname(lastname = 'Güney', middlename = 'Batı', firstname = 'Kuzey')  
#keyword argument
```

Kuzey Batı Güney

```
fullname('Güney')
```


TypeError Traceback (most recent call last)

<ipython-input-9-5894a4f19308> in <module>

----> 1 fullname('Güney')

TypeError: fullname() missing 2 required positional arguments:
'middlename' and 'lastname'

#Global scope

```
var1 = 100
```

```
def myfunc():  
    print(var1)
```

```
myfunc()  
print(var1)
```

100

100

#Local scope

```
def myfunc1():  
    var2 = 10  
    print(var2)
```

```
def myfunc2():  
    print(var2)
```

```
myfunc1()  
myfunc2()
```

```
10
```

```
-----  
-----
```

```
NameError                                Traceback (most recent call  
last)
```

```
<ipython-input-11-af329e9ac9fa> in <module>
```

```
8  
9 myfunc1()  
--> 10 myfunc2()
```

```
<ipython-input-11-af329e9ac9fa> in myfunc2()
```

```
5  
6 def myfunc2():  
----> 7     print(var2)  
8  
9 myfunc1()
```

```
NameError: name 'var2' is not defined
```

```
var1 = 100
```

```
def myfunc():  
    var1 = 99  
    print(var1)
```

```
myfunc()  
print(var1)
```

```
99  
100
```

```
#Fonksiyonlara argüman geçirme
```

```
list1 = [11, 22, 33, 44, 55]
```

```
def myfunc(list1):  
    del list1[0]
```

```
print('"List1" before calling the function :- ', list1)  
myfunc(list1)  
print('"List1" after calling the function :- ', list1)
```

```
"List1" before calling the function :- [11, 22, 33, 44, 55]
```

```
"List1" after calling the function :- [22, 33, 44, 55]
```

```

#Pass by reference
list1 = [11, 22, 33, 44, 55]

def myfunc(list1):
    list1.append(100)

print('"List1" before calling the function :- ', list1)
myfunc(list1)
print('"List1" after calling the function :- ', list1)

"List1" before calling the function :- [11, 22, 33, 44, 55]
"List1" after calling the function :- [11, 22, 33, 44, 55, 100]

list1 = [11, 22, 33, 44, 55]

def myfunc(list1):
    list1 = [10, 100, 1000, 10000]

print('"List1" before calling the function :- ', list1)
myfunc(list1)
print('"List1" after calling the function :- ', list1)

"List1" before calling the function :- [11, 22, 33, 44, 55]
"List1" after calling the function :- [11, 22, 33, 44, 55]

"""
def myfunc(positional arguments, *args (tuple), named (keyword)
arguments, **kwargs (dictionary))
def myfunc(120, 160, 50, b = 10, c = 20, a = 3)
"""

'\ndef myfunc(positional arguments, *args (tuple), named arguments,
**kwargs (dictionary))\ndef myfunc(120, 160, 50, b = 10, c = 20, a =
3\n'

def add(a,b,c):
    return a + b + c

add(10,20,30)

60

add(1,2,3,4)

-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-20-cd4c5f185a9e> in <module>
----> 1 add(1,2,3,4)

TypeError: add() takes 3 positional arguments but 4 were given

```

```

def some_args(arg_1, arg_2, arg_3):
    print("arg_1: ", arg_1)
    print("arg_2: ", arg_2)
    print("arg_3: ", arg_3)

my_list = [2, 3]
some_args(1, *my_list)

arg_1: 1
arg_2: 2
arg_3: 3

def add1(*args): ##*args argüman listesini dinamik olarak alabiliyor
    return sum(args)

print(add1(1,2,3))
print(add1(1,2,3,4))
print(add1(1,2,3,4,5))
print(add1(1,2,3,4,5,6))
print(add1(1,2,3,4,5,6,7))

6
10
15
21
28

list1 = [1,2,3,4,5,6,7]
tuple1 = (1,2,3,4,5,6,7)

add1(*list1), add1(*tuple1)

(28, 28)

list1 = [1,2,3,4,5,6,7]
list2 = [1,2,3,4,5,6,7]
list3 = [1,2,3,4,5,6,7]
list4 = [1,2,3,4,5,6,7]

add1(*list1, *list2, *list3, *list4)

112

def UserDetails(*args):
    print(args)

UserDetails('Doğu', 7412, 41102, 33, 'Türkiye', 'Turkish')

('Doğu', 7412, 41102, 33, 'Türkiye', 'Turkish')

def UserDetails(**kwargs):
    print(kwargs)

```

```

UserDetails(Name = 'Doğu', ID = 7412, Pincode = 41102, Age = 33,
Country = 'Türkiye', Language = 'Turkish')

{'Name': 'Doğu', 'ID': 7412, 'Pincode': 41102, 'Age': 33, 'Country':
'Türkiye', 'Language': 'Turkish'}

def UserDetails(**kwargs):
    for key, val in kwargs.items():
        print("{} :- {}".format(key, val))

UserDetails(Name = 'Doğu', ID = 7412, Pincode = 41102, Age = 33,
Country = 'Türkiye', Language = 'Turkish')

Name :- Doğu
ID :- 7412
Pincode :- 41102
Age :- 33
Country :- Türkiye
Language :- Turkish

mydict = {'Name': 'Doğu', 'ID': 7412, 'Pincode': 41102, 'Age': 33,
'Country': 'Türkiye', 'Language': 'Turkish'}

UserDetails(**mydict)

Name :- Doğu
ID :- 7412
Pincode :- 41102
Age :- 33
Country :- Türkiye
Language :- Turkish

def UserDetails(licenceNo, *args, phoneNo = 0, **kwargs):
    print('License No :- ', licenceNo)
    j = ''
    for i in args:
        j = j + i
    print('Full Name :- ', j)
    print('Phone Number :- ', phoneNo)
    for key, val in kwargs.items():
        print("{} :- {}".format(key, val))

name = ['Kuzey', ' ', 'Batı', ' ', 'Güney']
mydict = {'Name': 'Doğu', 'ID': 7412, 'Pincode': 41102, 'Age': 33,
'Country': 'Türkiye', 'Language': 'Turkish'}

UserDetails('TR145', *name, phoneNo = 1234567890, **mydict)

License No :- TR145
Full Name :- Kuzey Batı Güney

```

Phone Number :- 1234567890

Name :- Doğu

ID :- 7412

Pincode :- 41102

Age :- 33

Country :- Turkiye

Language :- Turkish

#default parameter

```
def printinfo(name, age = 23):
```

```
    print('Name :- ', name)
```

```
    print('Age :- ', age)
```

```
printinfo(age = 19, name = 'Tasmanian Devil')
```

```
printinfo(name = 'Bugs Bunny')
```

Name :- Tasmanian Devil

Age :- 19

Name :- Bugs Bunny

Age :- 23

```
import sys
```

```
print(sys.path)
```

```
['C:\\Users\\ITU', 'C:\\Users\\ITU\\anaconda3\\python38.zip', 'C:\\Users\\ITU\\anaconda3\\DLLs', 'C:\\Users\\ITU\\anaconda3\\lib', 'C:\\Users\\ITU\\anaconda3', '', 'C:\\Users\\ITU\\anaconda3\\lib\\site-packages', 'C:\\Users\\ITU\\anaconda3\\lib\\site-packages\\loket-0.2.1-py3.8.egg', 'C:\\Users\\ITU\\anaconda3\\lib\\site-packages\\win32', 'C:\\Users\\ITU\\anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\Users\\ITU\\anaconda3\\lib\\site-packages\\Pythonwin', 'C:\\Users\\ITU\\anaconda3\\lib\\site-packages\\IPython\\extensions', 'C:\\Users\\ITU\\.ipython']
```

```
from math import pi
```

```
print(pi)
```

3.141592653589793

```
import math
```

```
def area_of_circle(r):
```

```
    area = r**2 * math.pi
```

```
    return area
```

```
print(area_of_circle(15))
```

706.8583470577034

#Lambda, Map, Filter, Reduce (Anonymous Functions)

#lambda num1, num2 (arguments) : num1 + num2 (statement)

```

addition = lambda a : a + 10
print(addition(5))

15

product = lambda a, b : a * b
print(product(5, 6))

30

addition = lambda a, b, c : a + b + c
print(addition(5, 6, 2))

13

result = (lambda *args : sum(args))
result(10, 20), result(10, 20, 30, 40), result(10,20,30,40,50,60,70)

(30, 100, 280)

result1 = (lambda **kwargs : sum(kwargs.values()))
result1(a = 10, b = 20, c = 30), result1(a = 10, b = 20, c = 30, d =
40, e = 50)

(60, 150)

def product(nums): #user-defined func. ile anonim fonksiyon kullanımı
    total = 1
    for i in nums:
        total *= i #total = total * i
    return total

result1 = (lambda **kwargs : product(kwargs.values()))
result1(a = 10, b = 20, c = 30), result1(a = 10, b = 20, c = 30, d =
40, e = 50)

(6000, 120000000)

#filter -> filter(function, iterable (list, tuple vb.))

list1 = [1,2,3,4,5,6,7,8,9]

def odd(n):
    if n % 2 == 1:
        return True
    else:
        return False

odd_num = list(filter(odd, list1))
odd_num

[1, 3, 5, 7, 9]

```



```

list1 = [1,2,3,4,5,6,7,8,9]

odd_num = list(filter(lambda n : n%2 == 1, list1))
odd_num

[1, 3, 5, 7, 9]

#map -> map(function, iterable (list, tuple vb.))

def twice(n):
    return n*2

doubles = list(map(twice, odd_num))
doubles

[2, 6, 10, 14, 18]

doubles = list(map(lambda n : n*2, odd_num))
doubles

[2, 6, 10, 14, 18]

#reduce -> reduce(function, iterable (list, tuple vb.))
from functools import reduce

def add(a, b):
    return a + b

sum_all = reduce(add, doubles)
sum_all

50

sum_all = reduce(lambda a, b : a + b, doubles)
sum_all

50

sum_all = reduce(lambda a, b : a + b, list(map(lambda n : n*2,
list(filter(lambda n : n%2 == 1, list1)))))
sum_all

50

import operator

list2 = [1,2,3,4]

product = reduce(operator.mul, list2)

add = reduce(operator.add, list2)

concat_str = reduce(operator.add, ['Python', ' ', 'Rocks'])

```

```

prod = reduce(operator.mul, ['Hello ', 3])

min_num = reduce(lambda a, b : a if a < b else b, list2)

max_num = reduce(lambda a, b : a if a > b else b, list2)

print(product)
print(add)
print(concat_str)
print(prod)
print(min_num)
print(max_num)

24
10
Python Rocks
Hello Hello Hello
1
4

print(reduce(lambda a, b : bool(a and b), [0, 0, 1, 0, 0])) #False
print(reduce(lambda a, b : bool(a and b), [2, 3, 1, 5, 6])) #True
print(reduce(lambda a, b : bool(a and b), [8, 9, 1, 0, 9])) #False

print('-----')

print(reduce(lambda a, b : bool(a or b), [0, 0, 0, 0, 0])) #False
print(reduce(lambda a, b : bool(a or b), [2, 3, 1, 5, 6])) #True
print(reduce(lambda a, b : bool(a or b), [8, 9, 1, 0, 9])) #True

False
True
False
-----
False
True
True

#Faktöriyel
def factorial(num):
    if num <= 1: #base case
        return 1

    else: #recursive case
        return num * factorial(num - 1)

factorial(2950)

8714424896231723533236416994855066399501649438061404064312046054708657
5978025113094005316813387253233918824761027823220193224411432506537861

```

7356306517396908445480199457125827431878909595337211956991823884581123
1949402014722819099898252184822267366266441575822077543043363318103464
4940993940309924604620609902456032697423471842607702869889993931638206
1799349694007700953295759963400430503715311595884864409713551053373046
9771357581160746417987939320042460954121924979268801719874178741189533
2690770795383061042054819932275395875405966389713910757529276314956708
6400586167508612997192258532311723777113445816050221070943464596394722
2961539984442020693041144192668331964320149433156163079392550514494813
8965107042859934331312497135695163558053143484019469163773928148062796
9007628034685761551408460933042158339402568823136561302559358419861240
5388027312901059180735357625607204017341160020015631786472942665407686
2530283556842950559771181497121859669481388756581902669873576925704853
9888156890086292145922667553094572981826145313963816975826711373068430
1326190076001996449451445562236694839401215786191352691027992922711402
8389607761493678097685462315090776607713186695899388908807530473365169
8967633324169287889105666231929533809954890152525556887472299675700752
5712414808998141233843910301482486611585107860814140441192823648874202
6382712282520254324215957687520820919350400479553531931667795509970266
6830701655755599769380965227898081614989628498892402199723590903602491
9080417675158179023341703844258316778564375236214647788896734626583292
4825935280469297459611870569361765843847290444111864144303029098238954
1016596003896437989969323302415389676566567027537874205668250040870178
2522877260098521455849925612510095069425901915901779119521842687450345
9702469299590864658119839753677543377484462317338993831355286915196091
9313486266421894428221008309742655396435716843093679907885657254780858
3571325633571104610506318616058688404737492487679371145345149875298185
7801072684627064787718859626210820444254588084049291027402994449318170
3172622566934511315869012941244926229209949056086840379719435152210578
6896160692826004584907736233542607661826567964441925998462642526622266
2572306420673724612511283233331746672430990637081325591212188480074398
8150715642030549007045836083678210986396425203258752589790201041926171
0268570052400756719510155574203024133118574284737280069073973598283248
0007465327007688856088895540895957029861240077641396006951600952345755
1497062878668523124091926404366766229110816644212997304945467881898126
4391893543539094880524685800230308484550818659164216507988072789808747
5421918183228754363536770718523044943906220149274075958934769651113019
0187392609244680191139022620748791615597658162636307843455543215012038
2005887690316127909262061451169900248591934547727735370487670559303736
5380980371178139659512534680422455640847617201869529330458547713123454
0023829524029723436280249670228510566617489995448608221821296046201707
0885860336679901685781823147102544480026013419537049263959659484494775
7900212139366361303964804801833466245796281053051681448714745707705787
1469854858567921420197197690419420847130643969970741367736921283411196
2391140740062073306522695515688235607043620296860808426973741490705323
2343790331745651625454227579146910055621100198243766273431853984689693
7202405369816972914613713353661547584133079644066491528674874563071416
4439398763673696338195456012604922750379429642009810895522837629926347
8878968926409618388806650296400482035722790836006061382499466143374106
420593475334797772869698294241949730160060967540080679541619935580392

5126409829942803565210779891114430814607146761220487433293146044489146
1716608761312320316045627874930228095452330184399976943159663703564604
8877197809707959117627133011398355147741830134894890623142348696282548
7129221306949541035787627686247273102369977590260627549309933802759503
0685012849849633778759139135588082149222765206174152880443267999617573
7546973384690171228181510740735443120161707408089880916828072823702896
6611629378878196004643453329160317434747863389470670384348263337989106
0627038081943517899787955369262389872089699024952456269239195791470768
6963619976168277511538998912844089831744493721943968213700704305027376
4592715271181754137188976810273052604778654167691643024411805496017190
3046350205734127900802019906628638689537108246996197601013034536386090
9807724052245008129391658270880737472747437376650267631827902126162326
2147680627793929848925531813242360883150735836646467015477496317136205
1147784526823485104827256782409933093993576845642180629761137507532715
5582941944823780123095646909498496116173817609677963541399625351577052
5834429360725932781544972463411851358013679723237988978687753674345383
8033015989428675653728171343393314572382616459686525733677150340816333
5140981659581786538181218601600372009114244122523406040921940458007038
1149939474486333400081009851682800158362169177150891910742026096727243
5563598092955464414064580903105124117049312945562171166415473327706470
1432510931846216518868837886647621113989191111030677510244430750701227
6770579900638312453441837193539293274259997113640642320867003375018050
2750792876372772210054009458736955520049322432565594075483368897612722
5515785192244063244337906170955735270091366250126322856949074576768236
5011694450767070791349987261518997091964523491658760222827027755354851
0337552642355799519380279764994471461678869621694547922389299372905366
6062522976933502503616692879732205113292545065954297296471015404961782
0322820191571636047167495711140354403863678087215568759029641493265873
0888452056264965907888948004898495341511061683643177042227144225762932
9072188229129467277929786291100993610464503758902665655010118905648610
4174145519187421606908342790998800763545076379424983355422908203276371
2957019943517624074266088311391592000579476633333439201030782767735279
5011883047170192338286322600414758036309901138766433180322011642167036
9377299669318926290378696662720388334544938427246868614816206980395091
0108586371163899377050552656688929121026070982186894355540216232249313
2819349222911086960373487535532241113146116841287075983864688536073038
1294189248557422405770573042524344835446531408742562493299702019703806
9636489705229446101082995651239050541923185413340001051922469311801635
7269772536683515436241135998587200425643883886474012431494706349726295
6233553025865756937575366315175216687427388545148727421303213112042962
1710048525680326017469183209605443210905936285677796337125801859948018
6049183888069821135538934396794835930150035926415911201454183341534920
0968169256031604473140081301941622464060003728691928282233885035765191
0147796100769359017116644928381880903486194360463038295689951133530116
6840265832142671839738015427564619615258698361448059675001446021648988
2540511188027457003257267802965594091377059243393416941812243879109715
8268174736618608170613445415116204340901880140613472281164243309593784
3778405706828157315332746062275732848740065187781284626922819895239667
7668041473583819028151340296500715914150010404789816011230953544265481

9730969913810960030971588983739441860970891671528631504440772564979940
9399312606751808894955825740974795166192684467664876258960815749423186
3017056282604242807243429415186287647781479696983119573344401658337438
7902326179534348198789835619562783836790423967827710858523971508063126
5739775064829183588444292532082253796748093106616964230461167883918269
2622080209118836418059404464528979347600857007698809136966582255018793
3908701093841545514743155427620851907396567273131174092365871740445757
1935244348310263176153205522433016090758747040050675493349242947760545
2068237554070389890259116777157479465453752784778238982084981762529465
2142553007650414669320034693662071187059169559556537519727551763446976
7873651554369216034263105784885330296602368728269888723049823830582079
0242075141516332764362937335641352121145046818835117995987855286967209
3177926356603932464298683637219426038386631059195935818551311483139956
4684725989239374189469585562219317216466731957296618831558914011345517
1753141407460193231977123337544928113479487302390528812320845121236589
5104240364756318018532595997046496931917435491050828764135738029389642
8938179402108449224612148601456212716382126744024043861451234116301643
54542954624636702996586951606272000000000000000000000000000000000000
00
00
00
00
00
00
00
00
00
00
00
00
00

#n adet doğal sayının toplamı

```
def add(num):  
    if num == 0:  
        return 0  
  
    else:  
        return num + add(num - 1)
```

add(1589)

1263255

#fibonacci

```
def fibonacci(num):  
    if num <= 1:  
        return num  
    if num == 2:  
        return 1  
    else:  
        return(fibonacci(num - 1) + fibonacci(num - 2))
```

How many fibonacci numbers you want to generate? 50

[illegible]

```
<ipython-input-93-290f3adf2b58> in <module>
    10
    11 for i in range(nums):
--> 12     print(fibonacci(i))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
    5         return 1
    6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
    8
    9 nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
    5         return 1
    6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
    8
    9 nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
    5         return 1
    6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
    8
    9 nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
    5         return 1
    6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
    8
    9 nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
    5         return 1
    6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
    8
    9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9 nums = int(input("How many fibonacci numbers you want to generate? "))
```



```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```

5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9     nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9     nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9     nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9     nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
5         return 1
6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
8
9     nums = int(input("How many fibonacci numbers you want to
generate? "))

<ipython-input-93-290f3adf2b58> in fibonacci(num)
5         return 1
6     else:

```

```
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
      5         return 1
      6     else:
----> 7         return(fibonacci(num - 1) + fibonacci(num - 2))
      8
      9 nums = int(input("How many fibonacci numbers you want to
generate? "))
```

```
<ipython-input-93-290f3adf2b58> in fibonacci(num)
```

```
----> 1 def fibonacci(num):
      2     if num <= 1:
      3         return num
      4     if num == 2:
      5         return 1
```

KeyboardInterrupt:

#Files and Exceptions (Dosyalar ve İstisnalar)

```
fileobj = open('mydata\ibb_lojistik.txt', 'r')
fileobj.close()
```

```
fileobj = open('mydata\ibb_lojistik.txt', encoding="utf8")
fileobj.read()
```

"İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında \nen fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi\ntanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu\nilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu, \ndemiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir \nyerleşim dışı boş alanların fazlalığı gibi etmenler de önemli rol oynamaktadır. \nKullanılan alanların niteliğine bakıldığında ise, üretim alanlarının yoğunlaştığı \nilçeler yine yukarıda sayılan nedenlerden dolayı, Şile, Çatalca, Arnavutköy, \nSilivri, Beylikdüzü gibi yukarıda sayılan ilçelere benzer ilçelerdir. Depolama alanları \naçısından değerlendirildiğinde ise, açık depolama alanlarının Şile, Çatalca, Beylikdüzü, \nArnavutköy gibi ilçelerde yoğunlaştığı, kapalı depo alanlarının özellikle Ümraniye'de \naşırı derecede toplandığı, raflı depolama alanlarında Çekmeköy'ün yine aşırı derecede \nöne çıktığı, konteyner depolama alanlarında kapalı ve raflı depolama alanları kadar olmasa \nbile Beylikdüzü ve Eyüp ilçelerinin ön plana çıktığı, soğuk hava depolarında başta Silivri \nolmak üzere Esenyurt ve Başakşehir'in başta sayılabileceği, antrepolarda ise yine yoğunluk \nbeylikdüzü'nde olmak üzere Arnavutköy ve Sancaktepe'nin ön plana çıktığı görülmektedir(Tablo 3.6). \nSonuç olarak yük sevkiyatlarının giriş-çıkış noktalarının esas olarak depolama ve üretim alanları \nolduğu düşünülürse yukarıda adları geçen Ümraniye, Çatalca, Silivri, Arnavutköy, Şile, Ümraniye, \nbeylikdüzü, Çekmeköy gibi İstanbul'u bir anlamda çevreleyen ilçeler Lojistik Ana Planı hazırlanırken \ndikkatle analiz edilmesi gereken ilçelerin başında gelmektedir. Tablo 3.6'de genel olarak bahsedilen \nve ilçeler bazında ortaya çıkan görüntü, imalat sektöründeki ortalama alan kullanımının \nilçeler bazında dağılımında da aynı şekilde görülmektedir. Silivri, Çatalca, Arnavutköy, \nŞile, Büyükçekmece, Beylikdüzü gibi ilçeler ortalama alan kullanımında yoğunluğu çeken \nilçeler olmaktadır. Her ne kadar İstanbul ili içerisinde yerleşim alanları her ilçede yoğun \nbir şekilde görülmekteyse de merkeze en uzakta bulunan ve endüstriyel açıdan yoğunlaşmanın \ndaha fazla olduğu bu ilçelerde imalat sektörünün yoğun olarak görülmesi normaldir. \nBu yüzden lojistik açıdan değerlendirmeler yapılırken sevkiyat yoğunlukları, \nlojistik odak merkezleri, yükleme ve boşaltma alanları gibi unsurlar belirlenirken \nbu dağılım oldukça yol gösterici olacaktır. Özellikle İstanbul'a gerek Trakya ve Anadolu'dan \nve gerekse Avrupa ve Asya'dan kara, demir yada denizyolu ile gelişlerin \ntanbul'un sınır ilçeleri olarak kabul edilen bu ilçelerden gerçekleşiyor olması \nda İmalat Sektörü firmalarının buralarda yoğunlaşmasını daha net açıklamaktadır. \nBuna karşılık İstanbul'un nüfus ve yerleşim olarak yoğun ilçeleri olan Kadıköy, Bakırköy, \nÜsküdar, Fatih gibi ilçelerde ise doğal olarak imalat sektörü firmalarının daha az \nfaaliyet gösterdiği

görülmektedir.\n\nGünümüz yaşam koşullarının etkisiyle insan ve yük hareketliliği hızla artmaktadır. Kentsel\nyaşamdaki günlük ihtiyaçların karşılanması bakımından zorunlu hale gelen yük taşımacılığı,\n\ninsan yaşamı üzerinde çeşitli sosyal ve çevresel etkilere neden olmaktadır.\n\nÖzellikle büyük şehirlerde hızlı nüfus artışıyla ortaya çıkan ihtiyaçların, kısa sürelerde\ngiderilmesi çabası ulaştırma hizmetlerini zorlaştırmaktadır. Karayolu yolcu ve yük\ntaşımacılığında, son otuz yılda diğer türler ile yapılan taşımacılıklara kıyasla hızlı bir gelişme\n\nolmuş, bu dengesiz gelişmenin sonucu olarak karayolu taşımasının payı her iki taşıma için hızla\n\nartarak %90'ın üzerine çıkmıştır.\n\nÖzellikle İstanbul, sahip olduğu nüfus ve ekonomik potansiyel nedeniyle bu hızlı gelişimden\n\nçok etkilenen illerden biridir. Bu gerekçeyle; şehirdeki yük hareketlerinin oluşturduğu trafik\n\nsıkışıklıkları, erişim kısıtlamaları, park sorunları, emisyonlar gibi ekonomik, sosyal ve çevresel\n\netkiler göz önüne alınarak bir kentsel lojistik planlama yapılması ihtiyacı doğmuştur. Lojistik\n\nalanında yapılan yatırımlardan en fazla yararın elde edilmesi;\n\nAncak hedef ve amaçların iyi belirlenmesi,\n\nYapılan yatırımlar ile elde edilecek yararların maliyetlerle dengesinin sağlanması,\n\nYatırımları yapacak olan kuruluşların yatırım güçlerinin göz önünde tutulması,\n\nLojistik unsurlar içindeki entegrasyonun sağlanması,\n\nLojistik konusunda geleceğe yönelik taleplerin doğru kestirilebilmesine bağlıdır.\n\nDiğer yanda limanlar, demiryolları, karayolları ve bunların tesisleri ile depolama alanları kent\n\ncoğrafyası içinde yer bulmakta, kent içinde birçok kentsel kullanım ile birlikte yer almaktadır.\n\nDiğer kentsel kullanımlarla birlikte kent içinde yer alan lojistik yatırımların diğer kentsel\n\nkullanımlara ve kentsel yaşama olumsuz etkilerinin bertaraf edilmesi kapsamlı bir mekânsal\n\nplanlama ile sağlanabilir.\n\nBu amacın gerçekleşmesi ise ancak kapsamlı bir “Lojistik Ana Planı” ile mümkün\ngörülmektedir. Böyle bir Planın gerektirdiği araştırma, analiz, eylem planı ve değerlendirme\n\nraporuyla bu çalışmanın altyapısının oluşturulması hedeflenmektedir.\n\n1.1 Çalışmanın Amacı\n\nİstanbul Lojistik Ana Planı İşi Kapsamında Araç Sayımına Yönelik Saha Çalışması Yapılarak\n\nVeri Alımı Hizmeti İşinin amacı, İstanbul’un lojistik hareketliliğine dair elde edilecek verilerle\n\nİSTLAP’a bilimsel altlık oluşturmaktır. Çalışmada Teknik Şartname ile lokasyonları belirlenen\n\nlojistik odaklarda, perde-kordonlarda trafik sayımı ve anketler gerçekleştirilmiştir.\n\n2\n\nBu şekilde İstanbul’un lojistik altyapısının kente ve kentlilere olan sosyal, ekonomik, politik ve\nteknolojik etkileri belirlenerek tespit edilen sorunların çözümüne yönelik strateji ve öneriler\n\ngeliştirilecektir. Kısa, orta ve uzun dönemli oluşturulacak eylem planlarıyla bir yandan sağlıklı\n\nmekânsal gelişim sağlanırken diğer yandan İstanbul Ulaşım Ana Planı’na entegre bir gelişim\n\nsağlanacaktır.\n\nÇalışmada hızlı ve doğru bir veri kümesine ulaşmak için hazırlık aşamasından başlayarak veri\n\nnteslimine kadar olan süreçler önceden belirlenmiş metodoloji ve iş planına uygun şekilde\n\ngerçekleştirilmiştir.\n\n1.2 Çalışmanın Kapsamı\n\nÇalışma kapsamında, İstanbul’da yük taşıyan araç trafiği düzeyinin

belirlenerek lojistik atama\modelinde kullanılmak üzere, İstanbul, Gebze ve Dilovası'nda yer alan belirli lojistik odak ve\terminallerin yoğunlaştığı bölgelerin giriş- çıkışlarında, İstanbul trafiğini temsil eden perde\kordon hatlarında gerçekleştirilmiştir (Şekil 1.1).\nŞekil 1.1 Saha Çalışması Kapsamı\nPerde-Kordon Anketleri İstanbul Ulaşım Ana Planı çalışması kapsamında ele alınan perde\hatları noktalarında yapılmış olup sayımlar yük araçlarının yanı sıra tüm ulaşım araçları\ sayımlar da gerçekleştirilmiştir. Bu hat ve noktaların yanı sıra lojistik odak ve terminallerin\kapılarında sayım ve anket çalışmaları gerçekleştirilmiş, gelen giden yükün cinsi, doluluk\ndurumu, geldiği ve gittiği noktayı tespit etmeye yönelik veriler elde edilmiştir.\nÇalışma kapsamında aşağıda detaylı bilgileri verilen 33 lojistik odakta, 109 kesitte ve 5 dış\nistasyonda trafik sayımları ve anketler yapılmıştır. Teknik şartnamede 29 odak ve 99 kesit\nolarak belirtilmiştir.\n1. Lojistik Odak ve\Terminallerde Sayım ve Anket\n2. Perde-Kordonlarda Trafik\nSayımı\n3\n0daklarda yapılan sayımlarda şartnamede 5 kategori olarak belirtilen yük araçları 11\nekategoride sayılmıştır. Perde hatları ve dış istasyonlarda ise 5 tür lojistik yük araçlarına ek\nolarak 6 tür toplu ulaşım aracı olmak üzere toplam 11 kategoride sayım yapılmıştır"

```
fileobj.read()
```

```
''
```

```
fileobj.seek(0) #imleci başlangıç konumuna getirir  
fileobj.read()
```

"İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında \nen fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi\nİstanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu\nilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu, \ndemiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir \nyerleşim dışı boş alanların fazlalığı gibi etmenler de önemli rol oynamaktadır. \nKullanılan alanların niteliğine bakıldığında ise, üretim alanlarının yoğunlaştığı \nilçeler yine yukarıda sayılan nedenlerden dolayı, Şile, Çatalca, Arnavutköy, \nSilivri, Beylikdüzü gibi yukarıda sayılan ilçelere benzer ilçelerdir. Depolama alanları \naçısından değerlendirildiğinde ise, açık depolama alanlarının Şile, Çatalca, Beylikdüzü, \nArnavutköy gibi ilçelerde yoğunlaştığı, kapalı depo alanlarının özellikle Ümraniye'de \naşırı derecede toplandığı, raflı depolama alanlarında Çekmeköy'ün yine aşırı derecede \nöne çıktığı, konteyner depolama alanlarında kapalı ve raflı depolama alanları kadar olmasa \nbile Beylikdüzü ve Eyüp ilçelerinin ön plana çıktığı, soğuk hava depolarında başta Silivri \nolmak üzere Esenyurt ve Başakşehir'in başta sayılabileceği, antrepolarda ise yine yoğunluk \nBeylikdüzü'nde olmak üzere Arnavutköy ve Sancaktepe'nin ön plana çıktığı görülmektedir(Tablo 3.6). \nSonuç olarak yük sevkiyatlarının

giriş-çıkış noktalarının esas olarak depolama ve üretim alanları \nolduğu düşünülürse yukarıda adları geçen Ümraniye, Çatalca, Silivri, Arnavutköy, Şile, Ümraniye, \nBeylikdüzü, Çekmeköy gibi İstanbul'u bir anlamda çevreleyen ilçeler Lojistik Ana Planı hazırlanırken \ndikkatle analiz edilmesi gereken ilçelerin başında gelmektedir. Tablo 3.6'de genel olarak bahsedilen \nve ilçeler bazında ortaya çıkan görüntü, imalat sektöründeki ortalama alan kullanımının \nilçeler bazında dağılımında da aynı şekilde görülmektedir. Silivri, Çatalca, Arnavutköy, \nŞile, Büyükçekmece, Beylikdüzü gibi ilçeler ortalama alan kullanımında yoğunluğu çeken \nilçeler olmaktadır. Her ne kadar İstanbul ili içerisinde yerleşim alanları her ilçede yoğun \nbir şekilde görülmekteyse de merkeze en uzakta bulunan ve endüstriyel açıdan yoğunlaşmanın \ndaha fazla olduğu bu ilçelerde imalat sektörünün yoğun olarak görülmesi normaldir. \nBu yüzden lojistik açıdan değerlendirmeler yapılırken sevkiyat yoğunlukları, \nlojistik odak merkezleri, yükleme ve boşaltma alanları gibi unsurlar belirlenirken \nbu dağılım oldukça yol gösterici olacaktır. Özellikle İstanbul'a gerek Trakya ve Anadolu'dan \nve gerekse Avrupa ve Asya'dan kara, demir yada deniz yolu ile gelişlerin \nİstanbul'un sınır ilçeleri olarak kabul edilen bu ilçelerden gerçekleşiyor olması \nda İmalat Sektörü firmalarının buralarda yoğunlaşmasını daha net açıklamaktadır. \nBuna karşılık İstanbul'un nüfus ve yerleşim olarak yoğun ilçeleri olan Kadıköy, Bakırköy, \nÜsküdar, Fatih gibi ilçelerde ise doğal olarak imalat sektörü firmalarının daha az \nfaaliyet gösterdiği görülmektedir. \n\nGünümüz yaşam koşullarının etkisiyle insan ve yük hareketliliği hızla artmaktadır. Kentsel \nyaşamdaki günlük ihtiyaçların karşılanması bakımından zorunlu hale gelen yük taşımacılığı, \ninsan yaşamı üzerinde çeşitli sosyal ve çevresel etkilere neden olmaktadır. \nÖzellikle büyük şehirlerde hızlı nüfus artışıyla ortaya çıkan ihtiyaçların, kısa sürelerde \ngiderilmesi çabası ulaştırma hizmetlerini zorlaştırmaktadır. Karayolu yolcu ve yük \ntaşımacılığında, son otuz yılda diğer türler ile yapılan taşımacılıklara kıyasla hızlı bir gelişme \nolmuş, bu dengesiz gelişmenin sonucu olarak karayolu taşımasının payı her iki taşıma için hızla \nartarak %90'ın üzerine çıkmıştır. \nÖzellikle İstanbul, sahip olduğu nüfus ve ekonomik potansiyel nedeniyle bu hızlı gelişimden \nen çok etkilenen illerden biridir. Bu gerekçeyle; şehirdeki yük hareketlerinin oluşturduğu trafik \nsıkışıklıkları, erişim kısıtlamaları, park sorunları, emisyonlar gibi ekonomik, sosyal ve çevresel \netkiler göz önüne alınarak bir kentsel lojistik planlama yapılması ihtiyacı doğmuştur. Lojistik \nalanında yapılan yatırımlardan en fazla yararın elde edilmesi; \nAncak hedef ve amaçların iyi belirlenmesi, \nYapılan yatırımlar ile elde edilecek yararların maliyetlerle dengesinin sağlanması, \nYatırımları yapacak olan kuruluşların yatırım güçlerinin göz önünde tutulması, \nLojistik unsurlar içindeki entegrasyonun sağlanması, \nLojistik konusunda geleceğe yönelik taleplerin doğru kestirilebilmesine bağlıdır. \nDiğer yanda limanlar, demiryolları, karayolları ve bunların tesisleri ile

depolama alanları kent\ncöğrafyası içinde yer bulmakta, kent içinde birçok kentsel kullanım ile birlikte yer almaktadır.\nDiğer kentsel kullanımlarla birlikte kent içinde yer alan lojistik yatırımların diğer kentsel\nkullanımlara ve kentsel yaşama olumsuz etkilerinin bertaraf edilmesi kapsamlı bir mekânsal\ndanlama ile sağlanabilir.\nBu amacın gerçekleşmesi ise ancak kapsamlı bir "Lojistik Ana Planı" ile mümkün\ngörünmektedir. Böyle bir Planın gerektirdiği araştırma, analiz, eylem planı ve değerlendirme\nd raporuyla bu çalışmanın altyapısının oluşturulması hedeflenmektedir.\n1.1 Çalışmanın Amacı\nİstanbul Lojistik Ana Planı İş Kapsamında Araç Sayımına Yönelik Saha Çalışması Yapılarak\nVeri Alımı Hizmeti İşinin amacı, İstanbul'un lojistik hareketliliğine dair elde edilecek verilerle\nİSTLAP'a bilimsel altlık oluşturmaktır. Çalışmada Teknik Şartname ile lokasyonları belirlenen\nlojistik odaklarda, perde-kordonlarda trafik sayımı ve anketler gerçekleştirilmiştir.\n2\nBu şekilde İstanbul'un lojistik altyapısının kente ve kentlilere olan sosyal, ekonomik, politik ve\nteknolojik etkileri belirlenerek tespit edilen sorunların çözümüne yönelik strateji ve öneriler\ngeliştirilecektir. Kısa, orta ve uzun dönemli oluşturulacak eylem planlarıyla bir yandan sağlıklı\nd mekânsal gelişim sağlanırken diğer yandan İstanbul Ulaşım Ana Planı'na entegre bir gelişim\nsağlanacaktır.\nÇalışmada hızlı ve doğru bir veri kümesine ulaşmak için hazırlık aşamasından başlayarak veri\nd teslimine kadar olan süreçler önceden belirlenmiş metodoloji ve iş planına uygun şekilde\ngerçekleştirilmiştir.\n1.2 Çalışmanın Kapsamı\nÇalışma kapsamında, İstanbul'da yük taşıyan araç trafiği düzeyinin belirlenerek lojistik atama\nd modelinde kullanılmak üzere, İstanbul, Gebze ve Dilovası'nda yer alan belirli lojistik odak ve\nd terminallerin yoğunlaştığı bölgelerin giriş- çıkışlarında, İstanbul trafiğini temsil eden perde-kordon hatlarında gerçekleştirilmiştir (Şekil 1.1).\nŞekil 1.1 Saha Çalışması Kapsamı\nPerde-Kordon Anketleri İstanbul Ulaşım Ana Planı çalışması kapsamında ele alınan perde\nd hatları noktalarında yapılmış olup sayımlar yük araçlarının yanı sıra tüm ulaşım araçları\nd sayımlar da gerçekleştirilmiştir. Bu hat ve noktaların yanı sıra lojistik odak ve terminallerin\nd kapılarında sayım ve anket çalışmaları gerçekleştirilmiş, gelen giden yükün cinsi, doluluk\nd durumu, geldiği ve gittiği noktayı tespit etmeye yönelik veriler elde edilmiştir.\nÇalışma kapsamında aşağıda detaylı bilgileri verilen 33 lojistik odakta, 109 kesitte ve 5 dış\nd istasyonda trafik sayımları ve anketler yapılmıştır. Teknik şartnamede 29 odak ve 99 kesit\nd olarak belirtilmiştir.\n1. Lojistik Odak ve\nd Terminallerde Sayım ve Anket\n2. Perde-Kordonlarda Trafik\nd Sayımı\nd 3\nd Odaklarda yapılan sayımlarda şartnamede 5 kategori olarak belirtilen yük araçları 11\nd kategoride sayılmıştır. Perde hatları ve dış istasyonlarda ise 5 tür lojistik yük araçlarına ek\nd olarak 6 tür toplu ulaşım aracı olmak üzere toplam 11 kategoride sayım yapılmıştır"

```
fileobj.seek(7)
fileobj.read()
```


"er bazında firmaların kullandıkları ortalama alanlara bakıldığında \nen fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi\ntanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu\nilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu, \ndemiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir \nyerleşim dışı boş alanların fazlalığı gibi etmenler de önemli rol oynamaktadır. \nKullanılan alanların niteliğine bakıldığında ise, üretim alanlarının yoğunlaştığı \nilçeler yine yukarıda sayılan nedenlerden dolayı, Şile, Çatalca, Arnavutköy, \nSilivri, Beylikdüzü gibi yukarıda sayılan ilçelere benzer ilçelerdir. Depolama alanları \naçısından değerlendirildiğinde ise, açık depolama alanlarının Şile, Çatalca, Beylikdüzü, \nArnavutköy gibi ilçelerde yoğunlaştığı, kapalı depo alanlarının özellikle Ümraniye'de \naşırı derecede toplandığı, raflı depolama alanlarında Çekmeköy'ün yine aşırı derecede \nöne çıktığı, konteyner depolama alanlarında kapalı ve raflı depolama alanları kadar olmasa \nbile Beylikdüzü ve Eyüp ilçelerinin ön plana çıktığı, soğuk hava depolarında başta Silivri \nolmak üzere Esenyurt ve Başakşehir'in başta sayılabileceği, antrepolarda ise yine yoğunluk \nbeylikdüzü'nde olmak üzere Arnavutköy ve Sancaktepe'nin ön plana çıktığı görülmektedir(Tablo 3.6). \nSonuç olarak yük sevkiyatlarının giriş-çıkış noktalarının esas olarak depolama ve üretim alanları \nolduğu düşünülürse yukarıda adları geçen Ümraniye, Çatalca, Silivri, Arnavutköy, Şile, Ümraniye, \nbeylikdüzü, Çekmeköy gibi İstanbul'u bir anlamda çevreleyen ilçeler Lojistik Ana Planı hazırlanırken \ndikkatle analiz edilmesi gereken ilçelerin başında gelmektedir. Tablo 3.6'de genel olarak bahsedilen \nve ilçeler bazında ortaya çıkan görüntü, imalat sektöründeki ortalama alan kullanımının \nilçeler bazında dağılımında da aynı şekilde görülmektedir. Silivri, Çatalca, Arnavutköy, \nŞile, Büyükçekmece, Beylikdüzü gibi ilçeler ortalama alan kullanımında yoğunluğu çeken \nilçeler olmaktadır. Her ne kadar İstanbul ili içerisinde yerleşim alanları her ilçede yoğun \nbir şekilde görülmekteyse de merkeze en uzakta bulunan ve endüstriyel açıdan yoğunlaşmanın \ndaha fazla olduğu bu ilçelerde imalat sektörünün yoğun olarak görülmesi normaldir. \nBu yüzden lojistik açıdan değerlendirmeler yapılırken sevkiyat yoğunlukları, \nlojistik odak merkezleri, yükleme ve boşaltma alanları gibi unsurlar belirlenirken \nbu dağılım oldukça yol gösterici olacaktır. Özellikle İstanbul'a gerek Trakya ve Anadolu'dan \nve gerekse Avrupa ve Asya'dan kara, demir yada denizyolu ile gelişlerin \ntanbul'un sınır ilçeleri olarak kabul edilen bu ilçelerden gerçekleşiyor olması \nda İmalat Sektörü firmalarının buralarda yoğunlaşmasını daha net açıklamaktadır. \nBuna karşılık İstanbul'un nüfus ve yerleşim olarak yoğun ilçeleri olan Kadıköy, Bakırköy, \nÜsküdar, Fatih gibi ilçelerde ise doğal olarak imalat sektörü firmalarının daha az \nfaaliyet gösterdiği görülmektedir. \n\nGünümüz yaşam koşullarının etkisiyle insan ve yük hareketliliği hızla artmaktadır. Kentsel \nyaşamdaki günlük ihtiyaçların karşılanması bakımından zorunlu hale gelen yük

taşımacılığı, \ninsan yaşamı üzerinde çeşitli sosyal ve çevresel etkilere neden olmaktadır. \nÖzellikle büyük şehirlerde hızlı nüfus artışıyla ortaya çıkan ihtiyaçların, kısa sürelerde \ngiderilmesi çabası ulaştırma hizmetlerini zorlaştırmaktadır. Karayolu yolcu ve yük \ntaşımacılığında, son otuz yılda diğer türler ile yapılan taşımacılıklara kıyasla hızlı bir gelişme \nolmuş, bu dengesiz gelişmenin sonucu olarak karayolu taşımasının payı her iki taşıma için hızla \nartarak %90'ın üzerine çıkmıştır. \nÖzellikle İstanbul, sahip olduğu nüfus ve ekonomik potansiyel nedeniyle bu hızlı gelişimden \nen çok etkilenen illerden biridir. Bu gerekçeyle; şehirdeki yük hareketlerinin oluşturduğu trafik \nsıkışıklıkları, erişim kısıtlamaları, park sorunları, emisyonlar gibi ekonomik, sosyal ve çevresel \netkiler göz önüne alınarak bir kentsel lojistik planlama yapılması ihtiyacı doğmuştur. Lojistik \nalanında yapılan yatırımlardan en fazla yararın elde edilmesi; \nAncak hedef ve amaçların iyi belirlenmesi, \nYapılan yatırımlar ile elde edilecek yararların maliyetlerle dengesinin sağlanması, \nYatırımları yapacak olan kuruluşların yatırım güçlerinin göz önünde tutulması, \nLojistik unsurlar içindeki entegrasyonun sağlanması, \nLojistik konusunda geleceğe yönelik taleplerin doğru kestirilebilmesine bağlıdır. \nDiğer yanda limanlar, demiryolları, karayolları ve bunların tesisleri ile depolama alanları kent \ncoğrafyası içinde yer bulmakta, kent içinde birçok kentsel kullanım ile birlikte yer almaktadır. \nDiğer kentsel kullanımlarla birlikte kent içinde yer alan lojistik yatırımların diğer kentsel \nkullanımlara ve kentsel yaşama olumsuz etkilerinin bertaraf edilmesi kapsamlı bir mekânsal \nplanlama ile sağlanabilir. \nBu amacın gerçekleşmesi ise ancak kapsamlı bir "Lojistik Ana Planı" ile mümkün \ngörünmektedir. Böyle bir Planın gerektirdiği araştırma, analiz, eylem planı ve değerlendirme \nraporuyla bu çalışmanın altyapısının oluşturulması hedeflenmektedir. \n1.1 Çalışmanın Amacı \nİstanbul Lojistik Ana Planı İşi Kapsamında Araç Sayımına Yönelik Saha Çalışması Yapılarak \nVeri Alımı Hizmeti İşinin amacı, İstanbul'un lojistik hareketliliğine dair elde edilecek verilerle \nİSTLAP'a bilimsel altlık oluşturmaktır. Çalışmada Teknik Şartname ile lokasyonları belirlenen \nlojistik odaklarda, perde-kordonlarda trafik sayımı ve anketler gerçekleştirilmiştir. \n2 \nBu şekilde İstanbul'un lojistik altyapısının kente ve kentlilere olan sosyal, ekonomik, politik ve \nteknolojik etkileri belirlenerek tespit edilen sorunların çözümüne yönelik strateji ve öneriler \ngeliştirilecektir. Kısa, orta ve uzun dönemli oluşturulacak eylem planlarıyla bir yandan sağlıklı \nmekânsal gelişim sağlanırken diğer yandan İstanbul Ulaşım Ana Planı'na entegre bir gelişim \nsağlanacaktır. \nÇalışmada hızlı ve doğru bir veri kümesine ulaşmak için hazırlık aşamasından başlayarak veri \nteslimine kadar olan süreçler önceden belirlenmiş metodoloji ve iş planına uygun şekilde \ngerçekleştirilmiştir. \n1.2 Çalışmanın Kapsamı \nÇalışma kapsamında, İstanbul'da yük taşıyan araç trafiği düzeyinin belirlenerek lojistik atama \nmodelinde kullanılmak üzere, İstanbul, Gebze ve Dilovası'nda yer alan belirli lojistik odak ve \nterminallerin yoğunlaştığı bölgelerin giriş- çıkışlarında, İstanbul trafiğini temsil eden perdekordon hatlarında gerçekleştirilmiştir (Şekil 1.1). \nŞekil

1.1 Saha Çalışması Kapsamı\nPerde-Kordon Anketleri İstanbul Ulaşım Ana Planı çalışması kapsamında ele alınan perde\nhatları noktalarında yapılmış olup sayımlar yük araçlarının yanı sıra tüm ulaşım araçları\nsayımlar da gerçekleştirilmiştir. Bu hat ve noktaların yanı sıra lojistik odak ve terminallerin\nkapılarında sayım ve anket çalışmaları gerçekleştirilmiş, gelen giden yükün cinsi, doluluk\ndurumu, geldiği ve gittiği noktayı tespit etmeye yönelik veriler elde edilmiştir.\nÇalışma kapsamında aşağıda detaylı bilgileri verilen 33 lojistik odakta, 109 kesitte ve 5 dış\nistasyonda trafik sayımları ve anketler yapılmıştır. Teknik şartnamede 29 odak ve 99 kesit\nolarak belirtilmiştir.\n1. Lojistik Odak ve\nTerminallerde Sayım ve Anket\n2. Perde-Kordonlarda Trafik\nSayımı\n3. Odaklarda yapılan sayımlarda şartnamede 5 kategori olarak belirtilen yük araçları 11\nekategoride sayılmıştır. Perde hatları ve dış istasyonlarda ise 5 tür lojistik yük araçlarına ek\nolarak 6 tür toplu ulaşım aracı olmak üzere toplam 11 kategoride sayım yapılmıştır"

```
fileobj.seek(0)
```

```
fileobj.read(16)
```

```
'İlçeler bazında'
```

```
fileobj.seek(0)
```

```
fileobj.read(500)
```

```
'İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında \nen fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi\nİstanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu\nilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu, \ndemiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir \nyerleşim dışı boş alanların fazlalığı gibi etmenler de önemli rol oynamaktadır. \nKu'
```

```
fileobj.tell() #İmlecin güncel konumunu bize verir
```

```
566
```

```
fileobj.seek(0)
```

```
print(fileobj.readline())
```

```
print(fileobj.readline())
```

```
print(fileobj.readline())
```

```
print(fileobj.readline())
```

```
İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında
```

en fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi

İstanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu

ilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu,

```
fileobj.seek(0)
```

```
fileobj.readlines()
```

```
['İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında \n',  
 'en fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi\n',  
 'İstanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu\n',  
 'ilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu, \n',  
 'demiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir \n',  
 'yerleşim dışı boş alanların fazlalığı gibi etmenler de önemli rol oynamaktadır. \n',  
 'Kullanılan alanların niteliğine bakıldığında ise, üretim alanlarının yoğunlaştığı \n',  
 'ilçeler yine yukarıda sayılan nedenlerden dolayı, Şile, Çatalca, Arnavutköy, \n',  
 'Silivri, Beylikdüzü gibi yukarıda sayılan ilçelere benzer ilçelerdir. Depolama alanları \n',  
 'açısından değerlendirildiğinde ise, açık depolama alanlarının Şile, Çatalca, Beylikdüzü, \n',  
 'Arnavutköy gibi ilçelerde yoğunlaştığı, kapalı depo alanlarının özellikle Ümraniye'de \n',  
 'aşırı derecede toplandığı, raflı depolama alanlarında Çekmeköy'ün yine aşırı derecede \n',  
 'öne çıktığı, konteyner depolama alanlarında kapalı ve raflı depolama alanları kadar olmasa \n',  
 'bile Beylikdüzü ve Eyüp ilçelerinin ön plana çıktığı, soğuk hava depolarında başta Silivri \n',  
 'olmak üzere Esenyurt ve Başakşehir'in başta sayılabileceği, antrepolarda ise yine yoğunluk \n',  
 'Beylikdüzü'nde olmak üzere Arnavutköy ve Sancaktepe'nin ön plana çıktığı görülmektedir(Tablo 3.6). \n',  
 'Sonuç olarak yük sevkiyatlarının giriş-çıkış noktalarının esas olarak depolama ve üretim alanları \n',  
 'olduğu düşünülürse yukarıda adları geçen Ümraniye, Çatalca, Silivri,
```

Arnavutköy, Şile, Ümraniye, \n',
'Beylikdüzü, Çekmeköy gibi İstanbul'u bir anlamda çevreleyen ilçeler Lojistik Ana Planı hazırlanırken \n',
'dikkatle analiz edilmesi gereken ilçelerin başında gelmektedir. Tablo 3.6'de genel olarak bahsedilen \n',
've ilçeler bazında ortaya çıkan görüntü, imalat sektöründeki ortalama alan kullanımının \n',
'ilçeler bazında dağılımında da aynı şekilde görülmektedir. Silivri, Çatalca, Arnavutköy, \n',
'Şile, Büyükçekmece, Beylikdüzü gibi ilçeler ortalama alan kullanımında yoğunluğu çeken \n',
'ilçeler olmaktadır. Her ne kadar İstanbul ili içerisinde yerleşim alanları her ilçede yoğun \n',
'bir şekilde görülmekteyse de merkeze en uzakta bulunan ve endüstriyel açıdan yoğunlaşmanın \n',
'daha fazla olduğu bu ilçelerde imalat sektörünün yoğun olarak görülmesi normaldir. \n',
'Bu yüzden lojistik açıdan değerlendirmeler yapılırken sevkiyat yoğunlukları, \n',
'lojistik odak merkezleri, yükleme ve boşaltma alanları gibi unsurlar belirlenirken \n',
'bu dağılım oldukça yol gösterici olacaktır. Özellikle İstanbul'a gerek Trakya ve Anadolu'dan \n',
've gerekse Avrupa ve Asya'dan kara, demir yada deniz yolu ile gelişlerin \n',
'İstanbul'un sınır ilçeleri olarak kabul edilen bu ilçelerden gerçekleşiyor olması \n',
'da İmalat Sektörü firmalarının buralarda yoğunlaşmasını daha net açıklamaktadır. \n',
'Buna karşılık İstanbul'un nüfus ve yerleşim olarak yoğun ilçeleri olan Kadıköy, Bakırköy, \n',
'Üsküdar, Fatih gibi ilçelerde ise doğal olarak imalat sektörü firmalarının daha az \n',
'faaliyet gösterdiği görülmektedir. \n',
'\n',
'Günümüz yaşam koşullarının etkisiyle insan ve yük hareketliliği hızla artmaktadır. Kentsel \n',
'yaşamdaki günlük ihtiyaçların karşılanması bakımından zorunlu hale gelen yük taşımacılığı, \n',
'insan yaşamı üzerinde çeşitli sosyal ve çevresel etkilere neden olmaktadır. \n',
'Özellikle büyük şehirlerde hızlı nüfus artışıyla ortaya çıkan ihtiyaçların, kısa sürelerde \n',
'giderilmesi çabası ulaştırma hizmetlerini zorlaştırmaktadır. Karayolu yolcu ve yük \n',
'taşımacılığında, son otuz yılda diğer türler ile yapılan taşımacılıklara kıyasla hızlı bir gelişme \n',
'olmuş, bu dengesiz gelişmenin sonucu olarak karayolu taşımasının payı her iki taşıma için hızla \n',

"artarak %90'ın üzerine çıkmıştır.\n",
'Özellikle İstanbul, sahip olduğu nüfus ve ekonomik potansiyel nedeniyle bu hızlı gelişimden\n',
'en çok etkilenen illerden biridir. Bu gerekçeyle; şehirdeki yük hareketlerinin oluşturduğu trafik\n',
'sıkışıklıkları, erişim kısıtlamaları, park sorunları, emisyonlar gibi ekonomik, sosyal ve çevresel\n',
'etkiler göz önüne alınarak bir kentsel lojistik planlama yapılması ihtiyacı doğmuştur. Lojistik\n',
'alanında yapılan yatırımlardan en fazla yararın elde edilmesi;\n',
'\u00c0ncak hedef ve amaçların iyi belirlenmesi,\n',
'\u00c0 Yapılan yatırımlar ile elde edilecek yararların maliyetlerle dengesinin sağlanması,\n',
'\u00c0 Yatırımları yapacak olan kuruluşların yatırım güçlerinin göz önünde tutulması,\n',
'\u00c0 Lojistik unsurlar içindeki entegrasyonun sağlanması,\n',
'\u00c0 Lojistik konusunda geleceğe yönelik taleplerin doğru kestirilebilmesine bağlıdır.\n',
'Diğer yanda limanlar, demiryolları, karayolları ve bunların tesisleri ile depolama alanları kent\n',
'coğrafyası içinde yer bulmakta, kent içinde birçok kentsel kullanım ile birlikte yer almaktadır.\n',
'Diğer kentsel kullanımlarla birlikte kent içinde yer alan lojistik yatırımların diğer kentsel\n',
'kullanımlara ve kentsel yaşama olumsuz etkilerinin bertaraf edilmesi kapsamlı bir mekânsal\n',
'planlama ile sağlanabilir.\n',
'Bu amacın gerçekleşmesi ise ancak kapsamlı bir "Lojistik Ana Planı" ile mümkün\n',
'görülmektedir. Böyle bir Planın gerektirdiği araştırma, analiz, eylem planı ve değerlendirme\n',
'raporuyla bu çalışmanın altyapısının oluşturulması hedeflenmektedir.\n',
'1.1 Çalışmanın Amacı\n',
'İstanbul Lojistik Ana Planı İşİ Kapsamında Araç Sayımına Yönelik Saha Çalışması Yapılarak\n',
'Veri Alımı Hizmeti İşinin amacı, İstanbul'un lojistik hareketliliğine dair elde edilecek verilerle\n',
'İSTLAP'a bilimsel altlık oluşturmaktır. Çalışmada Teknik Şartname ile lokasyonları belirlenen\n',
'lojistik odaklarda, perde-kordonlarda trafik sayımı ve anketler gerçekleştirilmiştir.\n',
'2\n',
'Bu şekilde İstanbul'un lojistik altyapısının kente ve kentlilere olan sosyal, ekonomik, politik ve\n',
'teknolojik etkileri belirlenerek tespit edilen sorunların çözümüne yönelik strateji ve öneriler\n',
'geliştirilecektir. Kısa, orta ve uzun dönemli oluşturulacak eylem planlarıyla bir yandan sağlıklı\n',

```
'mekânsal gelişim sağlanırken diğer yandan İstanbul Ulaşım Ana
Planı'na entegre bir gelişim\n',
'sağlanacaktır.\n',
'Çalışmada hızlı ve doğru bir veri kümesine ulaşmak için hazırlık
aşamasından başlayarak veri\n',
'teslimine kadar olan süreçler önceden belirlenmiş metodoloji ve iş
planına uygun şekilde\n',
'gerçekleştirilmiştir.\n',
'1.2 Çalışmanın Kapsamı\n',
'Çalışma kapsamında, İstanbul'da yük taşıyan araç trafiği düzeyinin
belirlenerek lojistik atama\n',
'modelinde kullanılmak üzere, İstanbul, Gebze ve Dilovası'nda yer
alan belirli lojistik odak ve\n',
'terminallerin yoğunlaştığı bölgelerin giriş- çıkışlarında, İstanbul
trafiğini temsil eden perde-kordon hatlarında gerçekleştirilmiştir
(Şekil 1.1).\n',
'Şekil 1.1 Saha Çalışması Kapsamı\n',
'Perde-Kordon Anketleri İstanbul Ulaşım Ana Planı çalışması
kapsamında ele alınan perde\n',
'hatları noktalarında yapılmış olup sayımlar yük araçlarının yanı
sıra tüm ulaşım araçları\n',
'sayımlar da gerçekleştirilmiştir. Bu hat ve noktaların yanı sıra
lojistik odak ve terminallerin\n',
'kapılarında sayım ve anket çalışmaları gerçekleştirilmiş, gelen
giden yükün cinsi, doluluk\n',
'durumu, geldiği ve gittiği noktayı tespit etmeye yönelik veriler
elde edilmiştir.\n',
'Çalışma kapsamında aşağıda detaylı bilgileri verilen 33 lojistik
odakta, 109 kesitte ve 5 dış\n',
'istasyonda trafik sayımları ve anketler yapılmıştır. Teknik
şartnamede 29 odak ve 99 kesit\n',
'olarak belirtilmiştir.\n',
'1. Lojistik Odak ve\n',
'Terminallerde Sayım ve Anket\n',
'2. Perde-Kordonlarda Trafik\n',
'Sayımı\n',
'3\n',
'Odaklarda yapılan sayımlarda şartnamede 5 kategori olarak belirtilen
yük araçları 11\n',
'kategoride sayılmıştır. Perde hatları ve dış istasyonlarda ise 5 tür
lojistik yük araçlarına ek\n',
'olarak 6 tür toplu ulaşım aracı olmak üzere toplam 11 kategoride
sayım yapılmıştır']
```

```
fileobj.seek(0)
```

```
count = 0
```

```
for i in range(5):
    if(count < 5):
```

```
        print(fileobj.readline())
    else:
        break
    count += 1
```

İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında

en fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi

İstanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu

ilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu,

demiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir

```
fileobj = open('mydata\ibb_lojistik.txt', 'a', encoding = 'utf-8')
```

```
fileobj.write("THIS IS THE NEW CONTENT APPENDED IN THE FILE")
```

```
fileobj.close()
```

```
fileobj = open('mydata\ibb_lojistik.txt', encoding = 'utf-8')
```

```
fileobj.read()
```

"İlçeler bazında firmaların kullandıkları ortalama alanlara bakıldığında \nen fazla alan kullanımının Ümraniye, Çatalca, Silivri, Şile, Arnavutköy gibi\nİstanbul'un alan bakımından daha büyük ve yerleşim alanlarının daha kısıtlı olduğu\nilçeleri ön plana çıkmaktadır. Bunda arazi fiyatlarının daha düşük olması, karayolu, \ndemiryolu ve kısmen de olsa denizyolu ana bağlantı noktalarına yakınlık, kullanılabilir \nyerleşim dışı boş alanların fazlalığı gibi etmenler de önemli rol oynamaktadır. \nKullanılan alanların niteliğine bakıldığında ise, üretim alanlarının yoğunlaştığı \nilçeler yine yukarıda sayılan nedenlerden dolayı, Şile, Çatalca, Arnavutköy, \nSilivri, Beylikdüzü gibi yukarıda sayılan ilçelere benzer ilçelerdir. Depolama alanları \naçısından değerlendirildiğinde ise, açık depolama alanlarının Şile, Çatalca, Beylikdüzü, \nArnavutköy gibi ilçelerde yoğunlaştığı, kapalı depo alanlarının özellikle Ümraniye'de \naşırı derecede toplandığı, raflı depolama alanlarında Çekmeköy'ün yine aşırı derecede \nöne çıktığı, konteyner depolama alanlarında kapalı ve raflı depolama alanları kadar olmasa \nbile Beylikdüzü ve Eyüp ilçelerinin ön plana çıktığı, soğuk hava depolarında başta Silivri \nolmak üzere Esenyurt ve Başakşehir'in başta sayılabileceği, antrepolarda ise yine yoğunluk \n

nBeylikdüzü'nde olmak üzere Arnavutköy ve Sancaktepe'nin ön plana çıktığı görülmektedir (Tablo 3.6). \nSonuç olarak yük sevkiyatlarının giriş-çıkış noktalarının esas olarak depolama ve üretim alanları \n olduğu düşünülürse yukarıda adları geçen Ümraniye, Çatalca, Silivri, Arnavutköy, Şile, Ümraniye, \nBeylikdüzü, Çekmeköy gibi İstanbul'u bir anlamda çevreleyen ilçeler Lojistik Ana Planı hazırlanırken \ndikkatle analiz edilmesi gereken ilçelerin başında gelmektedir. Tablo 3.6'de genel olarak bahsedilen \nve ilçeler bazında ortaya çıkan görüntü, imalat sektöründeki ortalama alan kullanımının \nilçeler bazında dağılımında da aynı şekilde görülmektedir. Silivri, Çatalca, Arnavutköy, \nŞile, Büyükçekmece, Beylikdüzü gibi ilçeler ortalama alan kullanımında yoğunluğu çeken \nilçeler olmaktadır. Her ne kadar İstanbul ili içerisinde yerleşim alanları her ilçede yoğun \nbir şekilde görülmekteyse de merkeze en uzakta bulunan ve endüstriyel açıdan yoğunlaşmanın \ndaha fazla olduğu bu ilçelerde imalat sektörünün yoğun olarak görülmesi normaldir. \nBu yüzden lojistik açıdan değerlendirmeler yapılırken sevkiyat yoğunlukları, \nlojistik odak merkezleri, yükleme ve boşaltma alanları gibi unsurlar belirlenirken \nbu dağılım oldukça yol gösterici olacaktır. Özellikle İstanbul'a gerek Trakya ve Anadolu'dan \nve gerekse Avrupa ve Asya'dan kara, demir yada deniz yolu ile gelişlerin \nİstanbul'un sınır ilçeleri olarak kabul edilen bu ilçelerden gerçekleşiyor olması \nda İmalat Sektörü firmalarının buralarda yoğunlaşmasını daha net açıklamaktadır. \nBuna karşılık İstanbul'un nüfus ve yerleşim olarak yoğun ilçeleri olan Kadıköy, Bakırköy, \nÜsküdar, Fatih gibi ilçelerde ise doğal olarak imalat sektörü firmalarının daha az \nfaaliyet gösterdiği görülmektedir. \n\nGünümüz yaşam koşullarının etkisiyle insan ve yük hareketliliği hızla artmaktadır. Kentsel \nyaşamdaki günlük ihtiyaçların karşılanması bakımından zorunlu hale gelen yük taşımacılığı, \ninsan yaşamı üzerinde çeşitli sosyal ve çevresel etkilere neden olmaktadır. \nÖzellikle büyük şehirlerde hızlı nüfus artışıyla ortaya çıkan ihtiyaçların, kısa sürelerde \ngiderilmesi çabası ulaştırma hizmetlerini zorlaştırmaktadır. Karayolu yolcu ve yük \ntaşımacılığında, son otuz yılda diğer türler ile yapılan taşımacılıklara kıyasla hızlı bir gelişme \nolmuş, bu dengesiz gelişmenin sonucu olarak karayolu taşımasının payı her iki taşıma için hızla \nartarak %90'ın üzerine çıkmıştır. \nÖzellikle İstanbul, sahip olduğu nüfus ve ekonomik potansiyel nedeniyle bu hızlı gelişimden \nen çok etkilenen illerden biridir. Bu gerekçeyle; şehirdeki yük hareketlerinin oluşturduğu trafik \nsıkışıklıkları, erişim kısıtlamaları, park sorunları, emisyonlar gibi ekonomik, sosyal ve çevresel \netkiler göz önüne alınarak bir kentsel lojistik planlama yapılması ihtiyacı doğmuştur. Lojistik \nalanında yapılan yatırımlardan en fazla yararın elde edilmesi; \nAncak hedef ve amaçların iyi belirlenmesi, \nYapılan yatırımlar ile elde edilecek yararların maliyetlerle dengesinin sağlanması, \nYatırımları yapacak olan kuruluşların yatırım güçlerinin göz önünde tutulması, \nLojistik unsurlar içindeki entegrasyonun sağlanması, \nLojistik konusunda

geleceğe yönelik taleplerin doğru kestirilebilmesine bağlıdır.\nDiğer yanda limanlar, demiryolları, karayolları ve bunların tesisleri ile depolama alanları kent\ngoğrafyası içinde yer bulmakta, kent içinde birçok kentsel kullanım ile birlikte yer almaktadır.\nDiğer kentsel kullanımlarla birlikte kent içinde yer alan lojistik yatırımların diğer kentsel\nkullanımlara ve kentsel yaşama olumsuz etkilerinin bertaraf edilmesi kapsamlı bir mekânsal\ndiplanlama ile sağlanabilir.\nBu amacın gerçekleşmesi ise ancak kapsamlı bir “Lojistik Ana Planı” ile mümkün\ngörünmektedir. Böyle bir Planın gerektirdiği araştırma, analiz, eylem planı ve değerlendirme\nd raporuyla bu çalışmanın altyapısının oluşturulması hedeflenmektedir.\n1.1 Çalışmanın Amacı\nİstanbul Lojistik Ana Planı İş Kapsamında Araç Sayımına Yönelik Saha Çalışması Yapılarak\nVeri Alımı Hizmeti İşinin amacı, İstanbul’un lojistik hareketliliğine dair elde edilecek verilerle\nİSTLAP’a bilimsel altlık oluşturmaktır. Çalışmada Teknik Şartname ile lokasyonları belirlenen\nlojistik odaklarda, perde-kordonlarda trafik sayımı ve anketler gerçekleştirilmiştir.\n2\nBu şekilde İstanbul’un lojistik altyapısının kente ve kentlilere olan sosyal, ekonomik, politik ve\nteknolojik etkileri belirlenerek tespit edilen sorunların çözümüne yönelik strateji ve öneriler\ngeliştirilecektir. Kısa, orta ve uzun dönemli oluşturulacak eylem planlarıyla bir yandan sağlıklı\nd mekânsal gelişim sağlanırken diğer yandan İstanbul Ulaşım Ana Planı’na entegre bir gelişim\nsağlanacaktır.\nÇalışmada hızlı ve doğru bir veri kümesine ulaşmak için hazırlık aşamasından başlayarak veri\nd teslimine kadar olan süreçler önceden belirlenmiş metodoloji ve iş planına uygun şekilde\ngerçekleştirilmiştir.\n1.2 Çalışmanın Kapsamı\nÇalışma kapsamında, İstanbul’da yük taşıyan araç trafiği düzeyinin belirlenerek lojistik atama\nd modelinde kullanılmak üzere, İstanbul, Gebze ve Dilovası’nda yer alan belirli lojistik odak ve\nterminallerin yoğunlaştığı bölgelerin giriş- çıkışlarında, İstanbul trafiğini temsil eden perde-kordon hatlarında gerçekleştirilmiştir (Şekil 1.1).\nŞekil 1.1 Saha Çalışması Kapsamı\nPerde-Kordon Anketleri İstanbul Ulaşım Ana Planı çalışması kapsamında ele alınan perde\nd hatları noktalarında yapılmış olup sayımlar yük araçlarının yanı sıra tüm ulaşım araçları\nd sayımlar da gerçekleştirilmiştir. Bu hat ve noktaların yanı sıra lojistik odak ve terminallerin\nd kapılarında sayım ve anket çalışmaları gerçekleştirilmiş, gelen giden yükün cinsi, doluluk\nd durumu, geldiği ve gittiği noktayı tespit etmeye yönelik veriler elde edilmiştir.\nÇalışma kapsamında aşağıda detaylı bilgileri verilen 33 lojistik odakta, 109 kesitte ve 5 dış\nd istasyonda trafik sayımları ve anketler yapılmıştır. Teknik şartnamede 29 odak ve 99 kesit\nd olarak belirtilmiştir.\n1. Lojistik Odak ve\nd Terminallerde Sayım ve Anket\n2. Perde-Kordonlarda Trafik\nd Sayımı\n3. Odaklarda yapılan sayımlarda şartnamede 5 kategori olarak belirtilen yük araçları 11\nd kategoride sayılmıştır. Perde hatları ve dış istasyonlarda ise 5 tür lojistik yük araçlarına ek\nd olarak 6 tür toplu ulaşım aracı olmak üzere toplam 11 kategoride sayım yapılmıştır\nTHIS IS THE NEW CONTENT APPENDED IN THE FILE"

```

fileobj = open('mydata\libb_lojistik.txt', 'w', encoding = 'utf-8')

fileobj.write("NEW CONTENT ADDED IN THE FILE. PREVIOUS CONTENT HAS
BEEN OVERWRITTEN")

fileobj.close()

fileobj = open('mydata\libb_lojistik.txt', encoding = 'utf-8')

fileobj.read()

'NEW CONTENT ADDED IN THE FILE. PREVIOUS CONTENT HAS BEEN OVERWRITTEN'

fileobj = open("test5.txt", "w")

fileobj.write("First Line\n")
fileobj.write("Second Line\n")
fileobj.write("Third Line\n")
fileobj.write("Fourth Line\n")
fileobj.write("Fifth Line\n")
fileobj.close()

fileobj = open("test5.txt")

fileobj.readlines()

['First Line\n',
 'Second Line\n',
 'Third Line\n',
 'Fourth Line\n',
 'Fifth Line\n']

import os #operating system

os.remove('test2.txt')

os.remove('test2.txt')

-----
-----
FileNotFoundError                                Traceback (most recent call
last)
<ipython-input-10-1b9c7d4a7058> in <module>
----> 1 os.remove('test2.txt')

FileNotFoundError: [WinError 2] Sistem belirtilen dosyayı bulamıyor:
'test2.txt'

os.rmdir('folder1/') #Dizin (Directory) silme

import numpy as np

```

```

p5 = np.loadtxt('mydata\ibb_lojistik.txt', dtype = 'str')
p5
array(['NEW', 'CONTENT', 'ADDED', 'IN', 'THE', 'FILE.', 'PREVIOUS',
      'CONTENT', 'HAS', 'BEEN', 'OVERWRITTEN'], dtype='<U11')

p10 = np.genfromtxt('mydata\ibb_lojistik.txt', dtype = 'str')
p10
array(['NEW', 'CONTENT', 'ADDED', 'IN', 'THE', 'FILE.', 'PREVIOUS',
      'CONTENT', 'HAS', 'BEEN', 'OVERWRITTEN'], dtype='<U11')

#Hata ve İstisna İşleme (Error and Exception Handling)
"""
ArithmeticError, ZeroDivisionError, EOFError, IndexError, KeyError,
SyntaxError, IndentationError,
FileNotFoundError, try-except-else-finally
"""

import sys

try:
    print(100/0)

except:
    print(sys.exc_info()[1], 'exception occurred')

else:
    print('No exception occurred')

finally:
    print('Run this block of code always')

division by zero exception occurred
Run this block of code always

try:
    print(x)

except:
    print('Variable x is not defined')

Variable x is not defined

try:
    os.remove("test3.txt")

except:
    print('BELOW EXCEPTION OCCURED')
    print(sys.exc_info()[1])

```

```

else:
    print('No exception occurred')

finally:
    print('Run this block of code always')

BELOW EXCEPTION OCCURED
[WinError 2] Sistem belirtilen dosyayı bulamıyor: 'test3.txt'
Run this block of code always

try:
    x = int(input("Enter first number :- "))
    y = int(input("Enter second number :- "))
    print(x / y)
    os.remove("test3.txt")

except NameError:
    print("NameError exception occurred")

except FileNotFoundError:
    print("FileNotFoundError occurred")

except ZeroDivisionError:
    print("ZeroDivisionError occurred")

Enter first number :- 12
Enter second number :- 22
0.5454545454545454
FileNotFoundError occurred

try:
    x = int(input('Enter a number :- '))
    if x > 50:
        raise ValueError(x)

except:
    print(sys.exc_info()[0])

Enter a number :- 100
<class 'ValueError'>

try: #built-in exceptions
    import MyModule

except ModuleNotFoundError:
    print("ModuleNotFoundError exception occurred")

ModuleNotFoundError exception occurred

try:
    mydict = {1 : 'Doğu', 2: 'Batı', 3: 'Kuzey'}

```

```

    print(mydict[4])
except KeyError:
    print('KeyError exception raised')
KeyError exception raised

try:
    mylist = [1,2,3,4,5,6]
    print(mylist[10])
except IndexError:
    print("IndexError exception raised")
IndexError exception raised

#TypeError -> iki farklı veri tipi bir araya gelir ise bu istisna oluşur.
try:
    a = 50
    b = 'Güney'
    c = a / b

except TypeError:
    print("TypeError exception raised")
TypeError exception raised

#Classes and Objects (Sınıflar ve Nesneler)
"""
class myclass:
    "Docstring"
    def __init__(self, var1, var2):
        self.var1 = var1
        self.var2 = var2
    ...
    ...
    ...
    def myfunc1(self):
        print(self.var1)
        print(self.var1)

    def myfunc2(self):
        ...
"""

class myclass:
    var1 = 10

```

```

obj1 = myclass()
print(obj1.var1)

10

class Employee:
    """Employee isminde bir sınıf yaratıyoruz"""
    def __init__(self, name, empid):
        self.name = name
        self.empid = empid

    def greet(self):
        print("Thanks for joining Turkey Banks Association
{}!!!".format(self.name))

empl = Employee("Doğu", 34163) #instance (örnek) oluşturma

print('Name :- ', empl.name)
print('Employee ID :- ', empl.empid)
empl.greet()

```

```

Name :- Doğu
Employee ID :- 34163
Thanks for joining Turkey Banks Association Doğu!!!

```

```

empl.name = 'Batı'
empl.name

```

```
'Batı'
```

```

del empl.empid
empl.empid

```

```

-----
-----
AttributeError                                Traceback (most recent call
last)
<ipython-input-33-3bd7df91569c> in <module>
      1 del empl.empid
----> 2 empl.empid

```

```
AttributeError: 'Employee' object has no attribute 'empid'
```

```

del empl
empl

```

```

-----
-----
NameError                                    Traceback (most recent call
last)
<ipython-input-34-0b43e47f34a0> in <module>

```

```
1 del emp1
----> 2 emp1
```

NameError: name 'emp1' is not defined

```
emp2 = Employee('Michael', 34162)
```

```
print('Name :- ', emp2.name)
print('Employee ID :- ', emp2.empid)
emp2.greet()
```

Name :- Michael

Employee ID :- 34162

Thanks for joining Turkey Banks Association Michael!!!

```
emp2.country = 'United States of America'
emp2.country
```

'United States of America'

```
#Inheritance (Miras)
"""
```

Mevcut (ilk oluşturduğumuz) sınıflara literatürde -> Super Class, Parent Class, Base Class

Sonradan türettiğimiz sınıflara ise literatürde -> Sub Class, Child Class, Derived Class

"IS A" ilişkisi vardır. Student is a person.

```
"""
```