



**T.C. İSTANBUL ÜNİVERSİTESİ FEN FAKÜLTESİ
YÖK - TEBİP ÜSTÜN BAŞARILILAR PROGRAMI**



LİSANS BİTİRME TEZİ

PYTHON İLE VERİ ŞİFRELEME VE DEŞİFRELEME

ALTUĞ BEYHAN

**DANIŞMAN
PROF. DR. GÜLÇİN ÇİVİ BİLİR, İ.T.Ü.**

MAYIS 2023 - İSTANBUL

ÖNSÖZ

Bu tezde, dünya çapında popüler bir programlama dili olan Python'ın kriptoloji uygulamalarıyla birlikte tanıtılması amaçlanmaktadır. Bu amaç doğrultusunda birinci bölümde Python programlama dilinin kısa bir tarihi, dünya genelindeki kullanım istatistikleri ve en çok kullanılan bazı tümleşik geliştirme ortamları hakkında bilgi verilmektedir. İkinci bölümde temel Python programlama bilgisi çeşitli kodlama uygulamaları ile ele alınmaktadır. Bu uygulamalar için yazılan kodlar, bir GitHub deposuna dahil edilerek tezde paylaşılmıştır. Üçüncü bölümde ise kriptolojiye ilişkin temel kavramlar açıklanmış ve örneklenmiştir. Ardından, Python programlama dili için tanımlanmış başlıca modül ve paketler tanıtılmış ve bu araçlar kullanılarak bazı şifreleme algoritmalarının Python ile uygulaması yapılmıştır. Tezin Ek bölümünde ise Türkçe alfabe için tanımlanmış ve AES gibi modern kriptoloji algoritmalarının Türkçe metinlere uygulanmasında kullanılan ASCII Code Page-857 tablosuna yer verilmiştir.

Anahtar kelimeler: Python, Kriptoloji, Sezar Algoritması, Gelişmiş Şifreleme Standardı (AES), NumPy, PyCryptodome, Cryptography, Hashlib, Güvenli Özет Algoritması (SHA).

TEZDE YER ALAN BÖLÜMLER

BÖLÜM 1. PYTHON PROGRAMLAMA DİLİ

BÖLÜM 2. PYTHON PROGRAMLAMANIN TEMELLERİ

2.1. Giriş	2.2. Değişkenler	2.3. Temel Veri Tipleri	2.4. İndisleme ve Dilimleme
2.5. Fonksiyonlar ve Metotlar		2.6. Operatörler	2.7. Koşullar ve Döngüler

BÖLÜM 3. KRIPTOLOJİ VE PYTHON

3.1. Kriptoloji	3.2. Modül, Paket ve Kütüphaneler
3.3. Başlıca Kriptoloji Modül ve Paketleri	3.4. Python ile Kriptoloji Örnekleri

BÖLÜM 4. TARTIŞMA VE SONUÇ

BÖLÜM 1. PYTHON PROGRAMLAMA DİLİ

1989 Aralık: Hollandalı programcı Guido van Rossum, daha önce Amsterdam, Hollanda'da "Ulusal Matematik ve Bilgisayar Bilimleri Araştırma Enstitüsü (CWI)" kapsamında geliştirilmesine katkıda bulunduğu ABC programlama dilinin bazı zayıflıklarını ortadan kaldırmak üzere Python programlama dilinin temellerini attı (Lutz, 1996).

1991 Şubat: Python'ın ilk sürümü "Python 0.9" yayınlandı (Python Developer's Guide).

Günümüze kadar birçok gönüllünün çalışmasıyla yeni sürümler ortaya çıktı.

Günümüzde Python 3.7'den önceki sürümler artık desteklenmemektedir.

Ekim 2022: Python'ın şu ana kadar yayınlanan en son sürümü "Python 3.11" yayınlandı (Python Developer's Guide).

Ekim 2023: "Python 3.12" sürümünün yayınlanması beklenmektedir (Python Developer's Guide).

Python Programlama Dilinin Özellikleri

Python programlama dili, aşağıdaki özelliklere sahiptir:

- Yüksek seviyeli bir dildir. Kullanıcılarla kolay bir programlama imkânı sunar.
- Platformdan bağımsızdır. Bir program, farklı işletim sistemlerinde kodların değiştirilmesine gerek duyulmadan direkt çalıştırılabilir.
- Sade, anlaşılır ve yüksek okunabilirliği olan bir söz dizimine sahiptir. Bir programcı, daha önce yazmış olduğu kodu rahat okuyabilir ve başka programcılar da kodu kolayca anlayabilir.
- Çeşitli işlemler için hazırlanmış birçok modül, paket ve kütüphane desteği sunar.

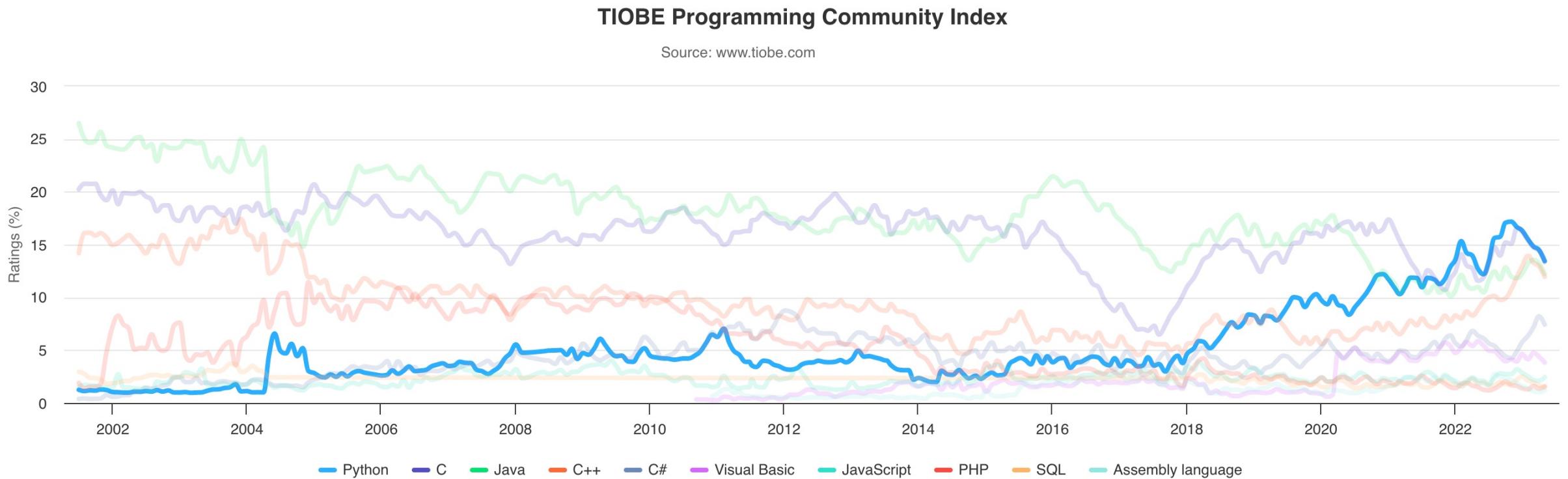
Dünyadaki En Popüler Programlama Dilleri

TIOBE Programlama Topluluğu İndeksi, her ay güncellenmekte ve programlama dillerinin dünya çapında kullanım oranlarını göstermektedir. Bu indekste derecelendirmeler, tüm dünyadaki yetenekli mühendislerin sayısı, programlama kursları ve üçüncü taraf satıcılar üzerinden yapılmakta ve Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube ve Baidu gibi popüler arama motorları ile hesaplanmaktadır. TIOBE Mayıs 2023 indeksine göre Python, %13,45 derece oranı ile dünyada en çok kullanılan programlama dili olarak 1. sırada yer almaktadır (TIOBE Index).

May 2023	May 2022	Change	Programming Language	Ratings	Change
1	1		 Python	13.45%	+0.71%
2	2		 C	13.35%	+1.76%
3	3		 Java	12.22%	+1.22%
4	4		 C++	11.96%	+3.13%
5	5		 C#	7.43%	+1.04%

Kaynak: TIOBE Index

Tüm Zamanlarda Python Kullanımı



Kaynak: TIOBE Index

Tümleşik Geliştirme Ortamları

Python programlamanın yapılabileceği birçok tümleşik geliştirme ortamı (IDE) bulunmaktadır. Bu ortamlardan bazıları

- IDLE (Python)
- PyCharm (JetBrains)
- Visual Studio Code (Microsoft)
- Jupyter Notebook (JupyterLab)

şeklinde verilebilir.

Tez kapsamında programlar PyCharm tümleşik geliştirme ortamında kodlanmıştır. PyCharm'ın en son sürümü <https://www.jetbrains.com/pycharm> adresinden indirilebilir.

```
import cipher
islem = True
while islem:
    metin = input("Metni giriniz (çıkış yapmak için boş bırakın): ")
    if metin == "":
        print("Çıkış yapıldı.")
        islem = False
    else:
        kontrol = input("Şifreleme işlemi için 1, deşifreleme işlemi için 2 giriniz: ")
        if kontrol == "1":
            x = cipher.Cipher()
            print("Şifreli metin: ",x.encrypt(metin),"\n")
        elif kontrol == "2":
            print("Deşifre edilmiş metin: ",x.decrypt(metin),"\n")
        else:
            print("Hatalı işlem girişi.\n")
```

Run: main
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
Metni giriniz (çıkış yapmak için boş bırakın): Altuğ Beyhan
Şifreleme işlemi için 1, deşifreleme işlemi için 2 giriniz: 1
Şifreli metin: ***@***@o+o@***@***@***

Metni giriniz (çıkış yapmak için boş bırakın): ***@***@o+o@***@***@***
Şifreleme işlemi için 1, deşifreleme işlemi için 2 giriniz: 2
Deşifre edilmiş metin: Altuğ Beyhan

Metni giriniz (çıkış yapmak için boş bırakın):
Çıkış yapıldı.
Process finished with exit code 0

2. PYTHON PROGRAMLAMANIN TEMELLERİ

The screenshot shows a GitHub repository page for 'LisansBitirmeTezi' containing a Jupyter Notebook file named 'AltugBeyhan_PythonileVeriSifrelemeveDesifreleme_TEBIPBitirmeTeziProgramlar.ipynb'. The notebook interface includes a sidebar for navigating files like 'main' and 'README.md'. The main area displays five code cells (In [2] to In [6]). Cell [2] prints 'Merhaba Dünya!'. Cell [3] prints a multi-line string. Cell [4] prints 'Merhaba Dünya!' followed by author information. Cell [5] defines a variable 'degisken' and prints its value. All outputs are displayed below their respective code snippets.

```
In [2]: print("Merhaba Dünya!")  
Merhaba Dünya!  
  
In [3]: #Merhaba Dünya Programı  
print("Merhaba Dünya!") #print: çıktı veren fonksiyondur  
#print("Bu yazı ekranда gözükmeyecek")  
print("Benim Adım Altuğ Beyhan.")  
  
Merhaba Dünya!  
Benim Adım Altuğ Beyhan.  
  
In [4]:  
"""  
Altuğ Beyhan  
2023  
www.altugbeyhan.com  
"""  
print("Merhaba Dünya!")  
  
Merhaba Dünya!  
  
In [5]: degisken = "Merhaba Dünya!"  
print(degisken)  
  
Merhaba Dünya!
```

Tezde yer alan tüm Python kodları açık kaynak olarak yayınlanmıştır:
<https://github.com/altugbeyhan/LisansBitirmeTezi>

2.1. Giriş

ekrana yazı
yazdırılan
fonksiyon

The screenshot shows a PyCharm interface with a code editor and a terminal window.

In the code editor (top half), the file `main.py` contains the following code:

```
print("Merhaba Dünya!")
```

Annotations with arrows point from the code to the terminal output:

- An arrow points from the word `print` to the terminal output, labeled "ekrana yazdırılan çıktı".
- An arrow points from the string "Merhaba Dünya!" to the terminal output, labeled "ekrana yazdırılacak girdi".

In the terminal window (bottom half), the command `/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py` is run, followed by the output:
`Merhaba Dünya!`
`Process finished with exit code 0`

2.2. Değişkenler

isimlendirilmiş geçici depolama konumları (Brookshear & Brylow, 2018)

değişken
adı

The screenshot shows a Python code editor in PyCharm with the file `main.py` open. The code contains two lines: `degisken = "Merhaba Dünya!"` and `print(degisken)`. A blue arrow points from the word `degisken` in the first line to the text `değişken adı`. Another blue arrow points from the assignment operator `=` to the text `atama operatörü`. A third blue arrow points from the string value `"Merhaba Dünya!"` to the text `değişkene atanmış veri`. Below the editor, the terminal window shows the output of running the script: `/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py`, followed by the printed message `Merhaba Dünya!`, and finally `Process finished with exit code 0`.

```
degisken = "Merhaba Dünya!"  
print(degisken)  
  
Process finished with exit code 0
```

2.3. Temel Veri Tipleri

The screenshot shows a PyCharm IDE interface. In the top editor window, the code `print(type("Merhaba Dünya!"))` is displayed. A blue arrow points from the word `type` in this code to the explanatory text "girdinin verisini döndüren fonksiyon". In the bottom terminal window, the command `/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python main.py` is run, followed by the output: `'` and "Process finished with exit code 0". A blue arrow points from the word `str` in the output to the explanatory text "girdi, \"str\" sınıfının bir nesnesidir yani girdinin veri tipi \"str\"dir".

```
PythonBitirmeTezi - main.py
Project: PythonBitirmeTezi
File: main.py
1 print(type("Merhaba Dünya!"))

Run: main
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
/Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
<class 'str'>
Process finished with exit code 0
```

girdinin verisini döndüren fonksiyon

girdi, "str" sınıfının bir nesnesidir
yani girdinin veri tipi "str"dir

2.3. Temel Veri Tipleri

VERİ TIPLERİ	AÇIKLAMA	VERİ ÖRNEKLERİ
str (string)	Karakter Dizisi	"Merhaba Dünya!"
int (integer)	Tam Sayı	17
float	Ondalıklı Sayı	3.14
complex	Karmaşık Sayı	3+4j
list	Liste	["altuğ", "beyhan", 22]
tuple	Demet	(8, 15, 17)
dict (dictionary)	Sözlük	{"elma":"apple", "kiraz":"cherry"}
set	Küme	{1, 2, 3, 4, 5}
bool (Boolean)	Doğru-Yanlış	True veya False

2.4. İndisleme ve Dilimleme

Bir dizinin belirli bir elemanına sıra numaralarıyla (indislerle) erişme işlemine indisleme, bir alt dizisine erişme işlemine ise dilimleme denir (Samancıoğlu, 2023).

Karakter dizisi	a	l	t	u	ğ
Sıra	1	2	3	4	5
İndis	0 / -5	1 / -4	2 / -3	3 / -2	4 / -1

İndisleme: veri_adi[indis_numarası]

Dilimleme: veri_adi[baslangic_indisi : bitis_indisi : atlama_miktari]

PythonBitirmeTezi – main.py

```
Project main.py
1 isim_soyisim = "Altuğ Beyhan"
2 kisilik_ozellikleri = ["Zeki", "Çalışkan", "Yardımcısever"]
3 kisel_bilgiler = ("Tekirdağlı", "22 Yaş", "Erkek", "Kriptografi")
4 print(isim_soyisim[0]) # 0. indis = 1. karakter
5 print(isim_soyisim[-1]) # -1. indis = sonuncu karakter
6 print(kisilik_ozellikleri[1]) # 1. indis = 2. eleman
7 print(kisilik_ozellikleri[-2]) # -2. indis = sondan 2. eleman
8 print(kisel_bilgiler[2]) # 2. indis = 3. eleman
9 print(kisel_bilgiler[-3]) # -3. indis = sondan 3. eleman
```

Run: main ×

```
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
/Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
A
n
Çalışkan
Çalışkan
Erkek
22 Yaş
```

Version Control Run Python Packages TODO Python Console Problems Terminal Services

PEP 8: W292 no newline at end of file

İndisleme

Dilimleme

PythonBitirmeTezi – main.py

```
Project main.py
1 isim = "altuğ beyhan"
2 sayılar = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
3 ünlüler = ("a", "e", "ı", "i", "o", "ö", "ü", "Ü")

4 print(isim[1:10:1]) # 1, 2, ..., 9. indisler (10. indis hariç)
5 print(sayılar[0:9:2]) # 0, 2, 4, 6, 8. indisler
6 print(unlüler[1:8:3]) # 1, 4, 7. indisler
```

Run: main ×

```
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
/Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
ltuğ beyh
[1, 3, 5, 7, 9]
('e', 'o', 'ü')
```

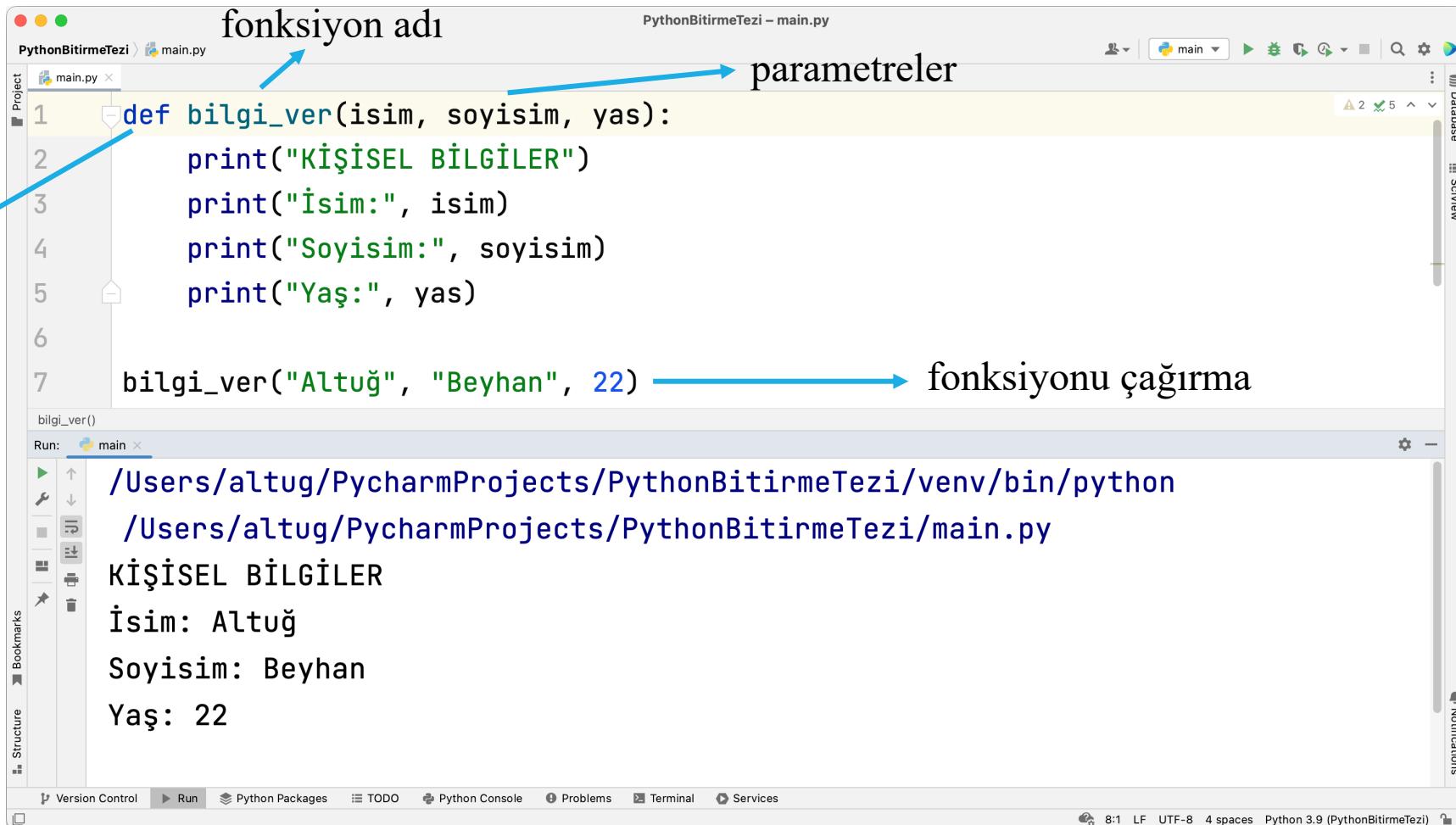
Version Control Run Python Packages TODO Python Console Problems Terminal Services

Notifications

2.5. Fonksiyonlar ve Metotlar

Belirli işlemlerle tanımlanan ve çağrırlığında bu işlemleri gerçekleştiren kod bloklarına fonksiyon adı verilir (Samancıoğlu, 2023).

fonksiyon
tanımlamamızı
sağlayan
söz dizimi



2.5. Fonksiyonlar ve Metotlar

"Sınıflar" içinde tanımlanan fonksiyonlara metot denir.
Metotlar `veri.metot_adi(parametreler)` şeklinde çağrılabılır (Samancıoğlu, 2023).

The screenshot shows the PyCharm IDE interface. The top window is titled "PythonBitirmeTezi – main.py". The code in the editor is:

```
listem = [1, 2, 3]
listem.append(4)
print(listem)
```

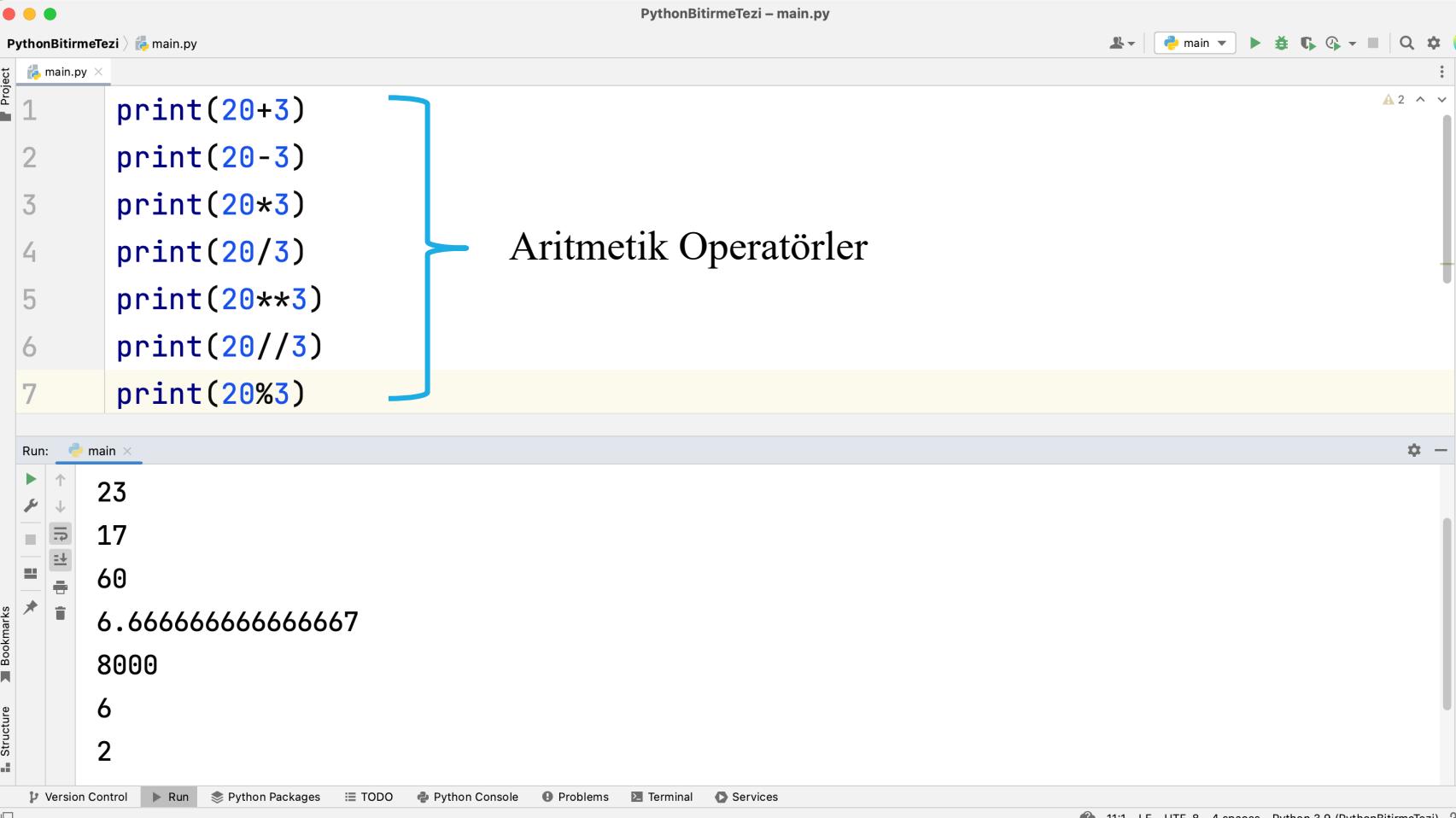
A blue bracket-shaped arrow points from the line "listem.append(4)" to the explanatory text below. The bottom window is titled "Run: main". The output shows the execution of the script:

```
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
/Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
[1, 2, 3, 4]
```

At the bottom, the message "Process finished with exit code 0" is displayed.

2.6. Operatörler

Python programlama dilinde, verilerin çeşitli işlemlere tabii tutulmasını sağlayan yapılara operatörler denir.



A screenshot of the PyCharm IDE interface. The main window shows a Python file named 'main.py' with the following code:

```
print(20+3)
print(20-3)
print(20*3)
print(20/3)
print(20**3)
print(20//3)
print(20%3)
```

The last line of code, `print(20%3)`, is highlighted with a yellow background. A blue curly brace on the left side of the code groups the first six lines under the heading "Aritmetik Operatörler". The output of the code is displayed in the "Run" tool window below, showing the results of each print statement:

```
23
17
60
6.666666666666667
8000
6
2
```

The PyCharm interface includes various toolbars and panels like Project, Run, Version Control, and Python Console.

Aritmetik Operatörler

2.7. Koşullar ve Döngüler

$x < 2$ ise

$2 \leq x < 10$ ise

aksi halde

Bir parçalı fonksiyonun koşullar ile tanımlanması örneği

The screenshot shows the PyCharm IDE interface. In the top navigation bar, it says "PythonBitirmeTezi - main.py". The code editor window contains the following Python code:

```
def f(x):
    if x < 2:
        return f"f({x}) = {-x**2}"
    elif 2 <= x < 10:
        return f"f({x}) = {x-2}"
    else:
        return f"f({x}) = {(x+6) ** (1/2)}"
print(f(-4), f(2), f(6), f(10), f(15), sep="\n")
```

To the right of the code, there is a mathematical formula box:

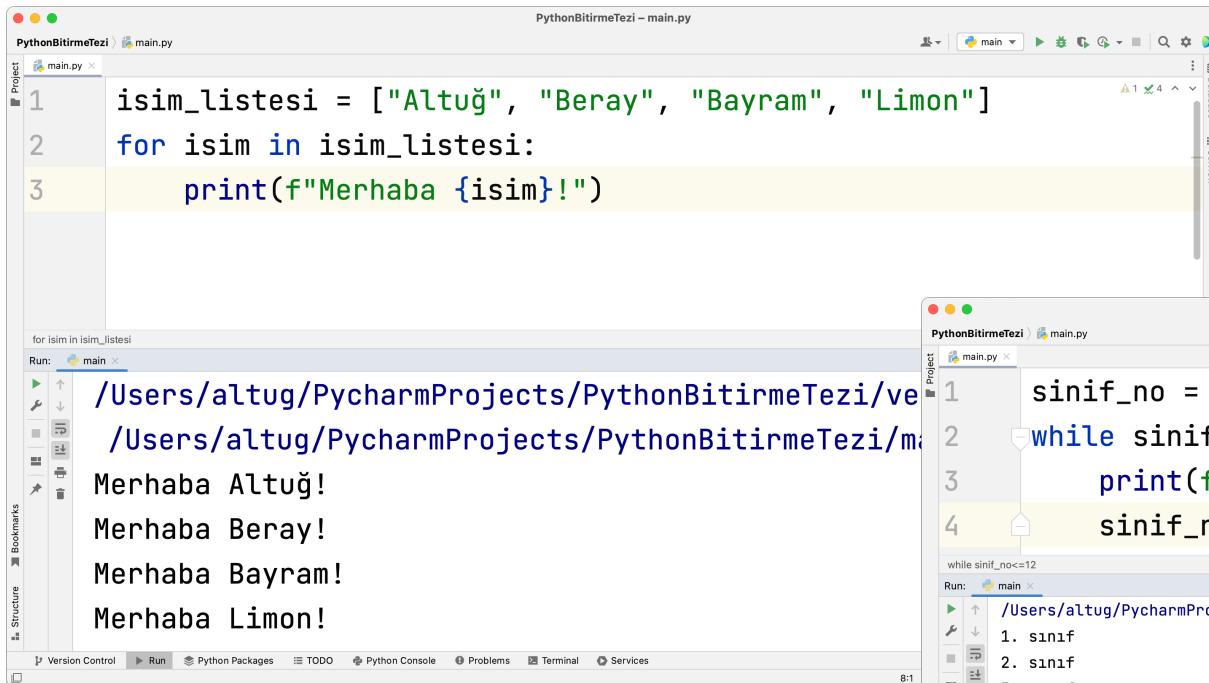
$$f : \mathbb{R} \rightarrow \mathbb{R}, \quad f(x) = \begin{cases} -x^2, & x < 2 \\ x - 2, & 2 \leq x < 10 \\ \sqrt{x + 6}, & x \geq 10 \end{cases}$$

The run output window at the bottom shows the results of the print statement:

```
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
/Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
f(-4) = -16
f(2) = 0
f(6) = 4
f(10) = 4.0
f(15) = 4.58257569495584
```

At the bottom of the PyCharm interface, there are several tabs: Version Control, Find, Run, Python Packages, TODO, Python Console, Problems, Terminal, and Services.

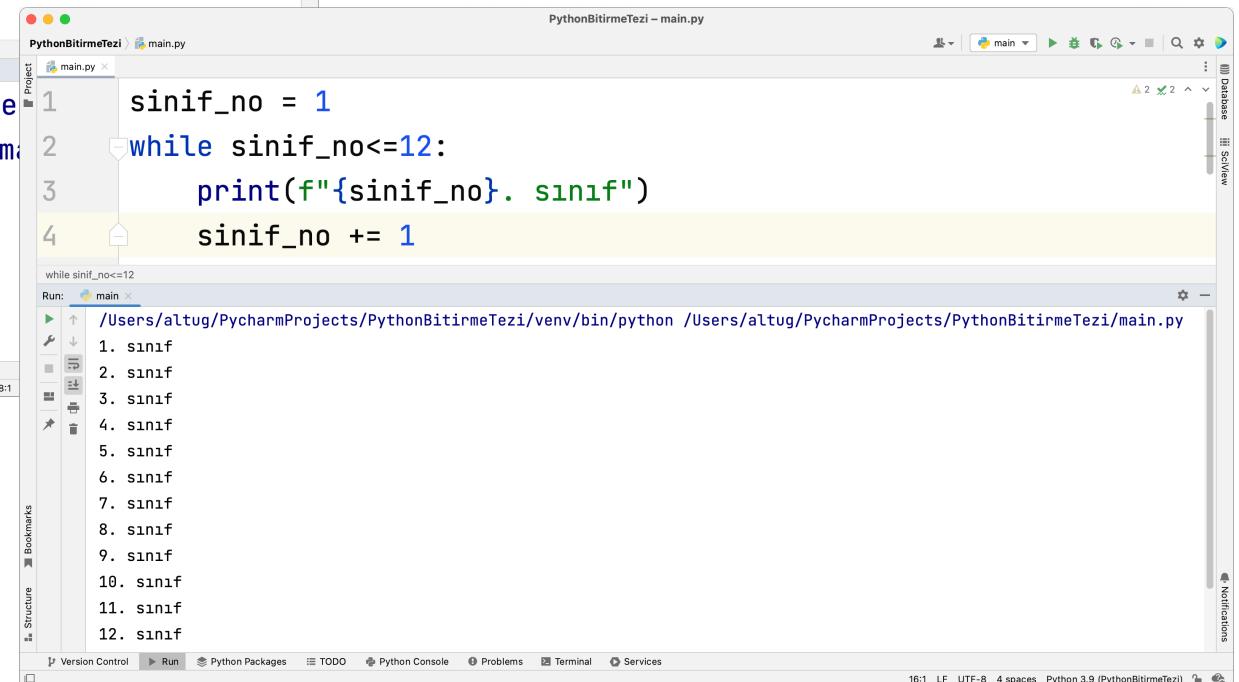
2.7. Koşullar ve Döngüler



```
PythonBitirmeTezi - main.py
Project main.py
1 isim_listesi = ["Altuğ", "Beray", "Bayram", "Limon"]
2 for isim in isim_listesi:
3     print(f"Merhaba {isim}!")

for isim in isim_listesi
Run: main
> /Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
Merhaba Altuğ!
Merhaba Beray!
Merhaba Bayram!
Merhaba Limon!
```

For döngüsü



```
PythonBitirmeTezi - main.py
Project main.py
1 sinif_no = 1
2 while sinif_no<=12:
3     print(f"{sinif_no}. sınıf")
4     sinif_no += 1

while sinif_no<=12
Run: main
> /Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
1. sınıf
2. sınıf
3. sınıf
4. sınıf
5. sınıf
6. sınıf
7. sınıf
8. sınıf
9. sınıf
10. sınıf
11. sınıf
12. sınıf
```

While döngüsü

BÖLÜM 3. KRİPTOLOJİ VE PYTHON



3.1. Kriptoloji

KAVRAM	AÇIKLAMA
Algoritma	Belli bir işin yapılması ve belli bir problemin çözülmesi için geliştirilen sistematik işlemler bütünüdür.
Şifreleme	Şifrelenmemiş verilerin şifreli hale getirilmesi işlemidir.
Deşifreleme	Şifrelenmiş verilerin şifresiz hale getirilmesi işlemidir.
Düz Metin	Şifreleme işlemi uygulanacak olan ve deşifreleme işlemi sonucu elde edilen orijinal metindir.
Şifreli Metin	Şifreleme işlemi sonucu elde edilen ve deşifreleme işlemi uygulanacak olan metindir.
Şifre	Verilerin şifrelenmesinde kullanılan algoritmalar ve kodlar bütünüdür.
Anahtar	Düz metinden şifreli metne veya şifreli metinden düz metne geçişte kullanılan simbol veya semboller dizisidir.

- **Şifre:** Sezar Algoritması
- **Düz Metin:** “BERAY”
- **Anahtar:** 3 (öteleme miktarı)
- **Şifreleme:**

B → C, Ç, **D**
E → F, G, **Ğ**
R → S, Ş, **T**
A → B, C, **Ç**
Y → Z, A, **B**

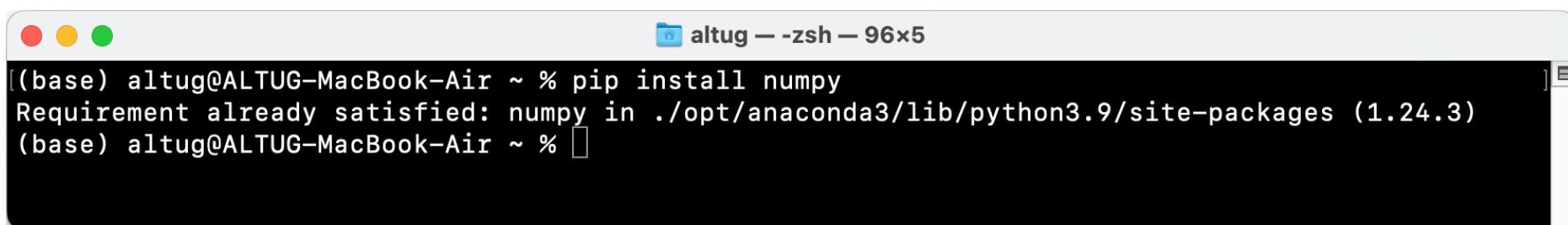
- **Şifreli Metin:** “DĞTÇB”
- **Deşifreleme:**

D → Ç, C, **B**
Ğ → G, F, **E**
T → Ş, S, **R**
Ç → C, B, **A**
B → A, Z, **Y**

3.2. Modül, Paket ve Kütüphaneler

Python programlama dilinde, içerisinde çeşitli kodları barındıran önceden kodlanmış proje dosyalarına modül adı verilir. Modüller bir araya gelerek paketleri, paketler bir araya gelerek kütüphaneleri oluşturur (Samancıoğlu, 2023).

Bir modülün, paketin veya kütüphanenin kullanılabilmesi için Python'da yüklü olması gerekmektedir. Bir modülü, paketi veya kütüphaneyi Python'a yüklemek için Komut İstemi/Terminal'de pip install <ad> komutu çalıştırılır (Samancıoğlu, 2023).

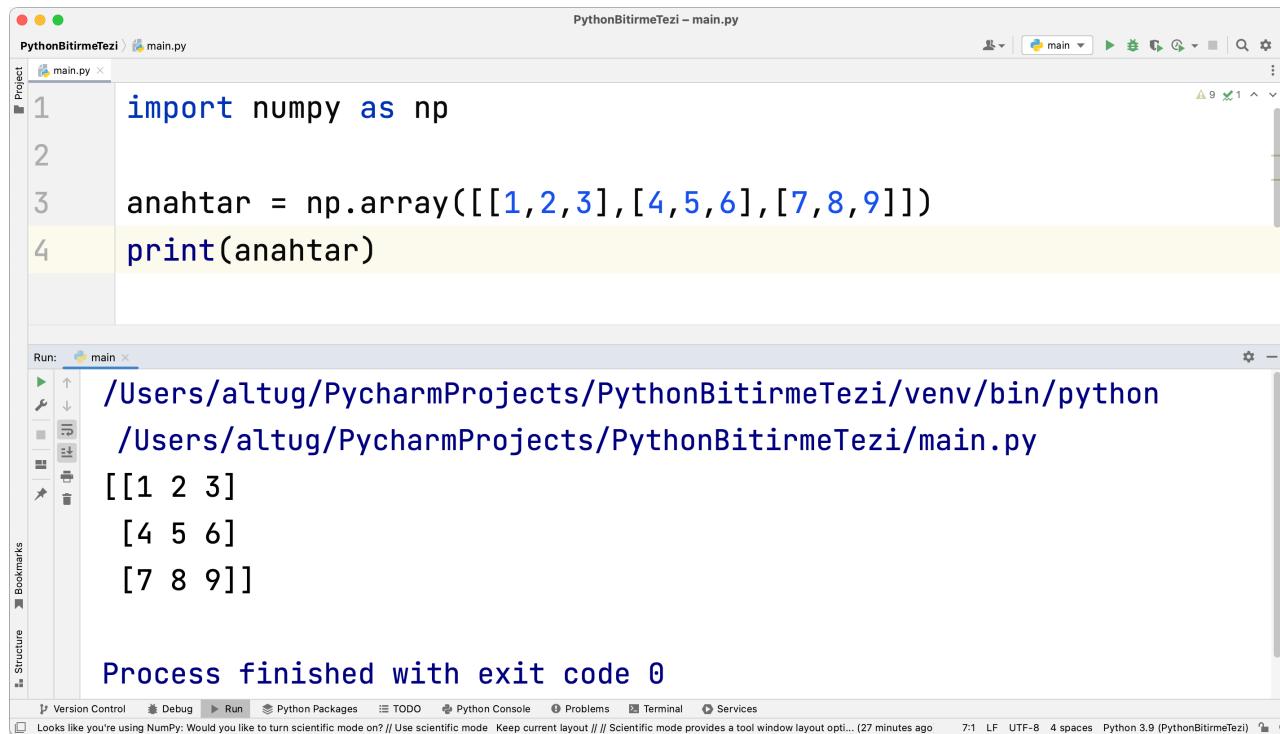


```
altug — -zsh — 96x5
(base) altug@ALTUG-MacBook-Air ~ % pip install numpy
Requirement already satisfied: numpy in ./opt/anaconda3/lib/python3.9/site-packages (1.24.3)
(base) altug@ALTUG-MacBook-Air ~ %
```

A screenshot of a macOS terminal window titled "altug — -zsh — 96x5". The window shows a command-line interface with a black background and white text. The user has run the command "pip install numpy". The output indicates that the requirement is already satisfied, showing the path to the numpy package in the Anaconda site-packages directory. The terminal window has the standard OS X title bar with red, yellow, and green buttons.

3.2. Modül, Paket ve Kütüphaneler

Bir proje dosyasında belirli bir modül, paket veya kütüphane içinde yer alan komutları kullanabilmek için modül, paket veya kütüphanenin proje dosyasına `import` söz dizimiyle dahil edilmesi gereklidir (Samancıoğlu, 2023).



The screenshot shows the PyCharm IDE interface. The top window displays the code in `main.py`:

```
PythonBitirmeTezi - main.py
Project: PythonBitirmeTezi
File: main.py
1 import numpy as np
2
3 anahtar = np.array([[1,2,3],[4,5,6],[7,8,9]])
4 print(anahtar)
```

The bottom window shows the run output:

```
Run: main
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
/Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

At the bottom, the message "Process finished with exit code 0" is displayed.

3.3. Başlıca Kriptoloji Modül ve Paketleri

3.3.1. PyCryptodome

“PyCryptodome”, düşük seviyeli ilkel kriptoloji algoritmalarını içeren bir Python paketidir (PyCryptodome Documentation).

- **Simetrik Şifreleme:** AES, DES, 3DES, CAST-128, RC2
- **Simetrik Blok Şifreleme Modları:** ECB, CBC, CFB, OFB, CTR, OpenPGP
- **Kimlik Doğrulamalı Şifreleme:** CCM, EAX, GCM, SIV, OCB, ChaCha20-Poly1305
- **Akiş Şifrelemesi:** Salsa20, ChaCha20, RC4
- **Kriptografik Özeti Fonksiyonları:** SHA-1, SHA-2, SHA-3, SHAKE128, SHAKE256, cSHAKE128, cSHAKE256, TupleHash128, TupleHash256, KangarooTwelve (XOF), Keccak, BLAKE2b, BLAKE2s, RIPE-MD160, MD5
- **Mesaj Doğrulama Kodları (MAC):** HMAC, CMAC, KMAC128, KMAC256, Poly1305
- **Asimetrik Anahtar Üretimi:** RSA, ECC (NIST P-eğrileri; Ed25519, Ed448), DSA, ElGamal
- **Asimetrik Anahtarlar için İçe ve Dışa Aktarma Formatı:** PEM, PKCS#8, ASN.1 DER
- **Asimetrik Şifreleme:** PKCS#1 (RSA)
- **Asimetrik Dijital İmzalar:** PKCS#1 (RSA), (EC)DSA, EdDSA
- **Anahtar Türetimi:** PBKDF2, scrypt, HKDF, PBKDF1
- **Diğer Kriptografik Protokoller:** Shamir Gizli Paylaşım, Blok Tamamlama

3.3. Başlıca Kriptoloji Modül ve Paketleri

3.3.2. Cryptography

“cryptography”, yüksek ve düşük seviyeli çeşitli kriptoloji algoritmalarını içeren bir Python paketidir. Pakette yüksek seviyeli algoritmalar “tarifler katmanı”, düşük seviyeli algoritmalar ise “tehlikeli maddeler katmanı” başlıkları altında toplanmıştır (Cryptography Documentation).

Yüksek seviyeli algoritmaların kullanımı kolay ve güvenlidir. Fakat düşük seviyeli algoritmaların kullanımı derin bir kriptografi bilgisini gerektirmekle birlikte, bu algoritmalar yapımcıları tarafından “tehlikeli maddeler” ifadesinin kısaltması ile ifade edilen “cryptography.hazmat” paketine yerleştirilmiştir (Cryptography Documentation).

- **Tarifler Katmanı**
 - Fernet (Simetrik Şifreleme)
 - X.509 Sertifikaları
- **Tehlikeli Maddeler Katmanı**
 - İlkeller
 - Kimlik Doğrulamalı Şifreleme
 - Asimetrik Algoritmalar
 - Sabit Zaman Fonksiyonları
 - Anahtar Türetim Fonksiyonları
 - Anahtar Paketleme
 - Mesaj Doğrulama Kodları
 - Özetleme
 - Simetrik Şifreleme
 - Simetrik Blok Tamamlama
 - Çift Faktörlü Kimlik Doğrulama
 - İstisnalar
 - Rastgele Sayı Üretimi

3.3. Başlıca Kriptoloji Modül ve Paketleri

3.3.3. Hashlib

“hashlib”, Python programlama dilinde yerleşik olarak gelen bir özet fonksiyonları modülüdür (Python Documentation).

MD5, SHA1, SHA224, SHA256, SHA384, SHA512, SHA3-224, SHA3-256, SHA3-384, SHA3-512, BLAKE2b, BLAKE2s, scrypt, PBKDF2

3.4. Python ile Kriptoloji Örnekleri

The screenshot shows the PyCharm IDE interface with a Python project named "PythonBitirmeTezi". The main.py file contains the following code:

```
def tekliyerkoyma():
    alfabe = "ABCÇDEFGĞHIİJKLMNOÖPRSŞTUÜYZ"
    sayac = 1
    while True:
        print("İŞLEMLER: Şifreleme ($)/Deşifreleme (D)/Kaba Kuvvet Saldırısı(K)/Çıkış (Ç)")
        karar = input(f"{sayac} numaralı işlemi seçiniz: ")

        if karar.upper() == "$": # Şifreleme işlemi
            duz_metin = input("Şifrelenecek metni giriniz: ").replace("i", "İ").upper()
            try:
                anahtar = int(input("Anahtar sayısını giriniz: ")) % len(alfabe)
            except:
                print("Lütfen geçerli bir sayı giriniz.\n")
                continue
            sifreli_metin = ""
            for i in range(len(duz_metin)):
                sifreli_metin += alfabe[(duz_metin[i] - 'A') + anahtar]
            print(f"Sifreli Metin: {sifreli_metin}")
        elif karar.upper() == "D":
            sifreli_metin = input("Şifrelenecek metni giriniz: ").upper()
            anahtar = int(input("Anahtar sayısını giriniz: "))
            duz_metin = ""
            for i in range(len(sifreli_metin)):
                duz_metin += alfabe[(sifreli_metin[i] - 'A') - anahtar]
            print(f"Düz Metin: {duz_metin}")
        elif karar.upper() == "K":
            sifreli_metin = input("Şifrelenecek metni giriniz: ").upper()
            anahtar = int(input("Anahtar sayısını giriniz: "))
            duz_metin = ""
            for i in range(len(sifreli_metin)):
                duz_metin += alfabe[(sifreli_metin[i] - 'A') - anahtar]
            print(f"Düz Metin: {duz_metin}")
        else:
            break
```

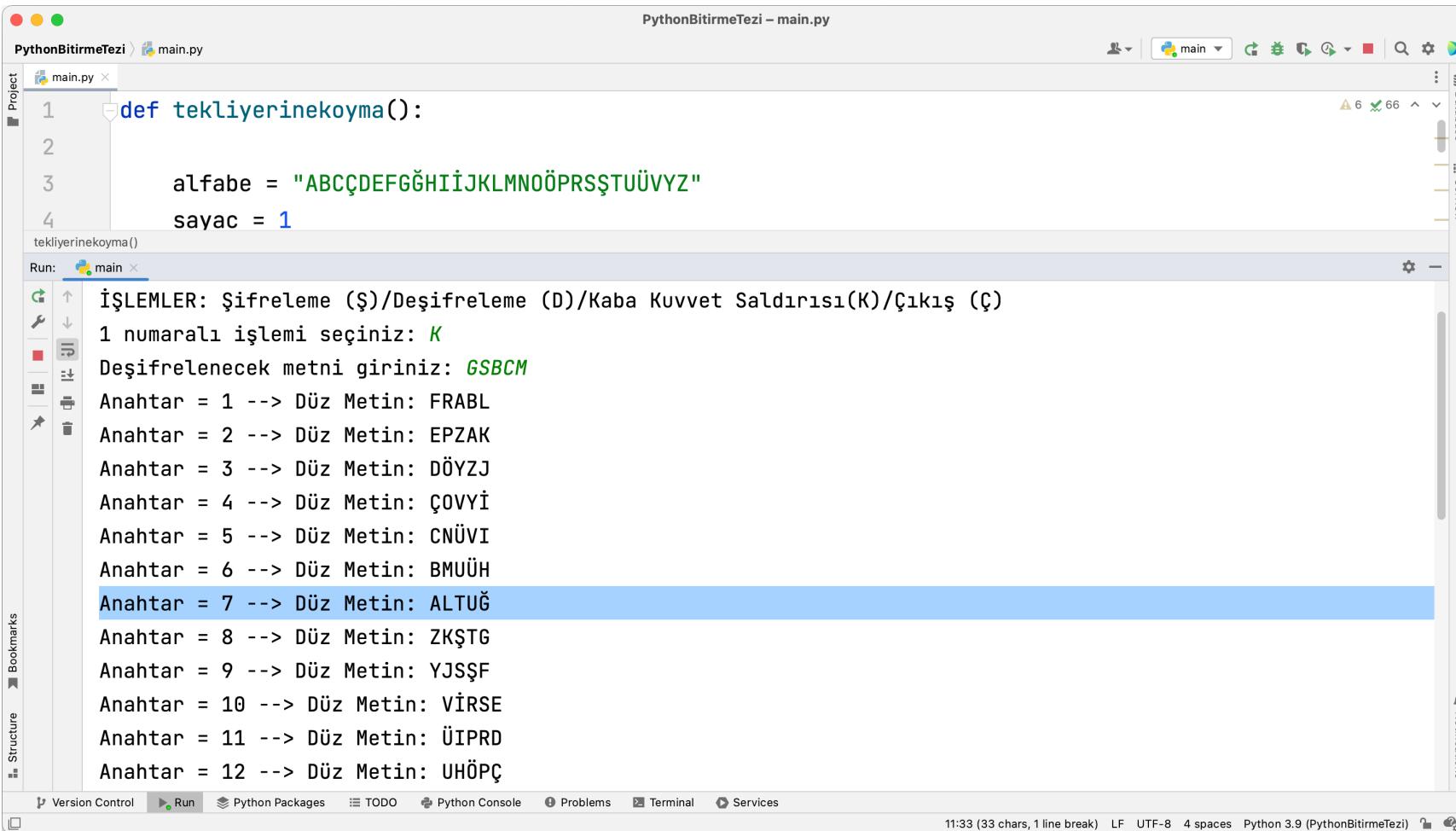
The Run tab shows the execution results:

```
İŞLEMLER: Şifreleme ($)/Deşifreleme (D)/Kaba Kuvvet Saldırısı(K)/Çıkış (Ç)
1 numaralı işlemi seçiniz: $
Şifrelenecek metni giriniz: ALTUĞ
Anahtar sayısını giriniz: 7
Sifreli Metin: GSBCM

İŞLEMLER: Şifreleme ($)/Deşifreleme (D)/Kaba Kuvvet Saldırısı(K)/Çıkış (Ç)
2 numaralı işlemi seçiniz: D
Deşifrelenecek metni giriniz: GSBCM
Anahtar sayısını giriniz: 7
Düz Metin: ALTUĞ
```

At the bottom, there is a message about NumPy scientific mode.

3.4. Python ile Kriptoloji Örnekleri



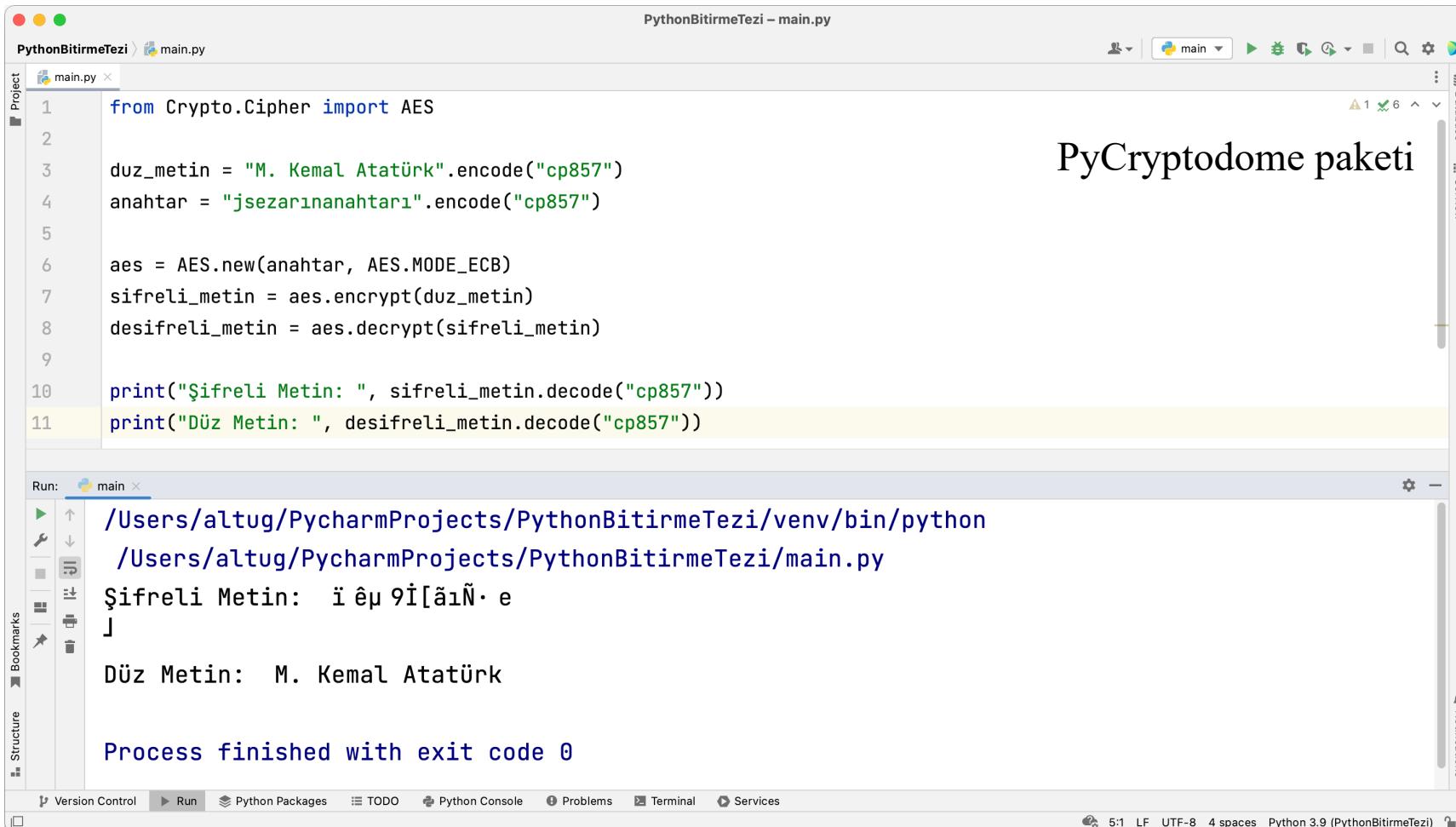
The screenshot shows a Python script named `main.py` in a PyCharm IDE. The code defines a function `tekliyerinekoyma()` which takes a string and applies a monoalphabetic substitution cipher using a specific key. The output shows various deciphered messages for different keys.

```
def tekliyerinekoyma():

    alfabe = "ABCÇDEFGĞHIİJKLMNOÖPRSŞTUÜVYZ"
    sayac = 1

    tekliyerinekoyma()
    Run: main ×
    İŞLEMLER: Şifreleme ($)/Deşifreleme (D)/Kaba Kuvvet Saldırısı(K)/Çıkış (Ç)
    1 numaralı işlemi seçiniz: K
    Deşifrelenecek metni giriniz: GSBCM
    Anahtar = 1 --> Düz Metin: FRABL
    Anahtar = 2 --> Düz Metin: EPZAK
    Anahtar = 3 --> Düz Metin: DÖYZJ
    Anahtar = 4 --> Düz Metin: ÇOVYİ
    Anahtar = 5 --> Düz Metin: CNÜVI
    Anahtar = 6 --> Düz Metin: BMUÜH
    Anahtar = 7 --> Düz Metin: ALTUĞ
    Anahtar = 8 --> Düz Metin: ZKŞTG
    Anahtar = 9 --> Düz Metin: YJSŞF
    Anahtar = 10 --> Düz Metin: VİRSE
    Anahtar = 11 --> Düz Metin: ÜİPRD
    Anahtar = 12 --> Düz Metin: UHÖPÇ
```

3.4. Python ile Kriptoloji Örnekleri



PyCryptodome paketi

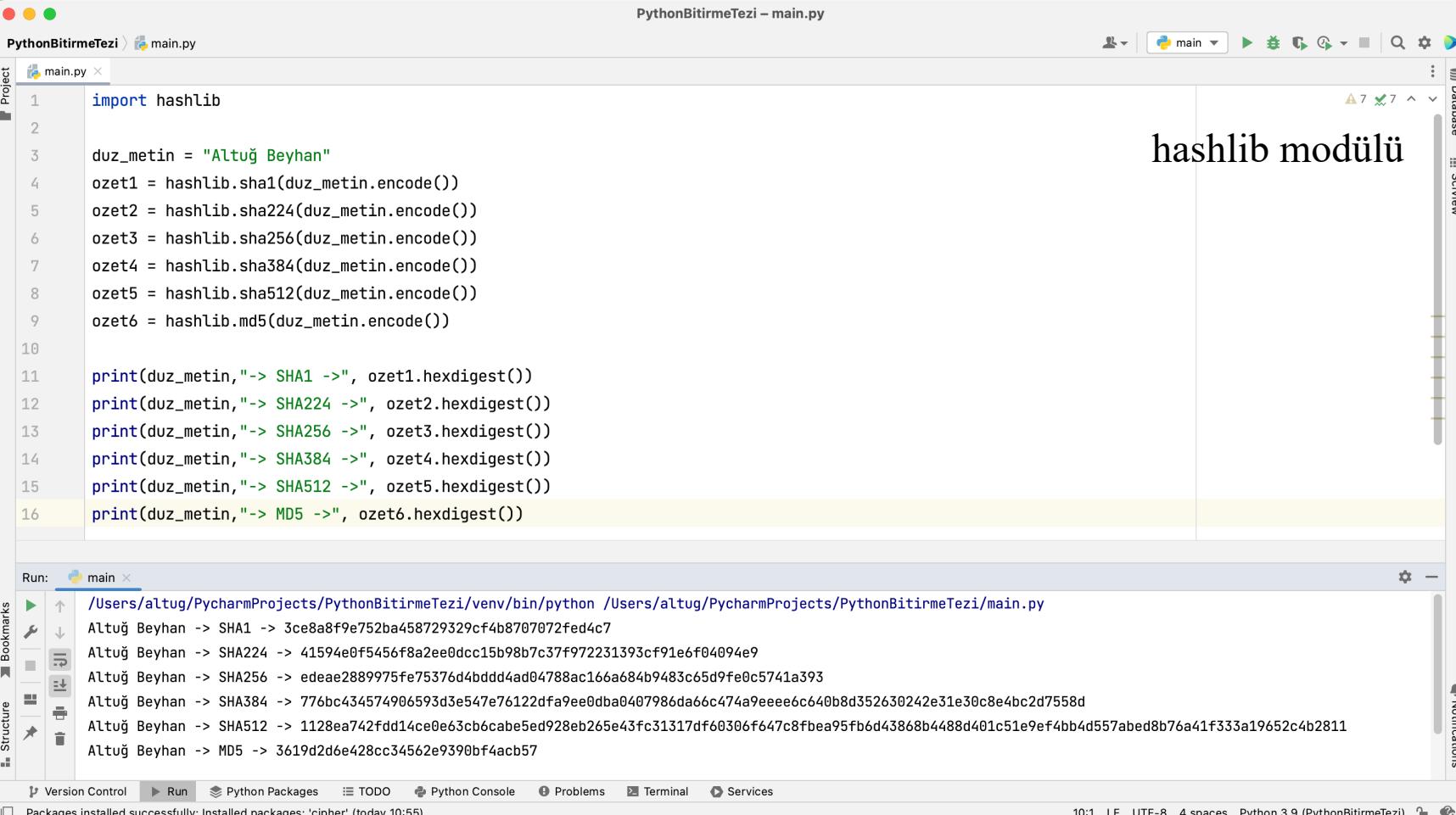
```
PythonBitirmeTezi - main.py
Project: PythonBitirmeTezi | File: main.py
1  from Crypto.Cipher import AES
2
3  duz_metin = "M. Kemal Atatürk".encode("cp857")
4  anahtar = "jsezarınanahtarı".encode("cp857")
5
6  aes = AES.new(anahtar, AES.MODE_ECB)
7  sifreli_metin = aes.encrypt(duz_metin)
8  desifreli_metin = aes.decrypt(sifreli_metin)
9
10 print("Şifreli Metin: ", sifreli_metin.decode("cp857"))
11 print("Düz Metin: ", desifreli_metin.decode("cp857"))

Run: main
Run: main
  /Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python
  /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py
  Şifreli Metin:  İ ēü 9İ[ãıÑ· e
  ]
  Düz Metin:  M. Kemal Atatürk

Process finished with exit code 0
```

The screenshot shows a PyCharm IDE interface with a Python script named 'main.py' open. The script uses the PyCryptodome library's AES module to encrypt and decrypt a string. The output window shows the encrypted and decrypted strings. The status bar at the bottom indicates the file is 5:1, LF, UTF-8, 4 spaces, and was run with Python 3.9.

3.4. Python ile Kriptoloji Örnekleri



The screenshot shows a PyCharm project titled "PythonBitirmeTezi" with a file named "main.py". The code uses the "hashlib" module to calculate SHA1, SHA224, SHA256, SHA384, SHA512, and MD5 hash digest values for the string "Altug Beyhan". The output of the run command shows the resulting hexdigest values for each algorithm.

```
import hashlib

duz_metin = "Altug Beyhan"

ozet1 = hashlib.sha1(duz_metin.encode())
ozet2 = hashlib.sha224(duz_metin.encode())
ozet3 = hashlib.sha256(duz_metin.encode())
ozet4 = hashlib.sha384(duz_metin.encode())
ozet5 = hashlib.sha512(duz_metin.encode())
ozet6 = hashlib.md5(duz_metin.encode())

print(duz_metin,"-> SHA1 ->", ozet1.hexdigest())
print(duz_metin,"-> SHA224 ->", ozet2.hexdigest())
print(duz_metin,"-> SHA256 ->", ozet3.hexdigest())
print(duz_metin,"-> SHA384 ->", ozet4.hexdigest())
print(duz_metin,"-> SHA512 ->", ozet5.hexdigest())
print(duz_metin,"-> MD5 ->", ozet6.hexdigest())
```

Run: main ×
/Users/altug/PycharmProjects/PythonBitirmeTezi/venv/bin/python /Users/altug/PycharmProjects/PythonBitirmeTezi/main.py

Altuğ Beyhan ->	SHA1 ->	SHA224 ->	SHA256 ->	SHA384 ->	SHA512 ->	MD5 ->
Altug Beyhan	3ce8a8f9e752ba458729329cf4b8707072fed4c7	41594e0f5456f8a2ee0dcc15b98b7c37f972231393cf91e6f04094e9	edae2889975fe75376d4bdd4ad04788ac166a684b9483c65d9fe0c5741a393	776bc434574906593d3e547e76122dfa9ee0dba0407986da66c474a9eee6c640b8d352630242e31e30c8e4bc2d7558d	1128ea742fd14ce0e63cb6cabef5ed928eb265e43fc31317df60306f647c8fbea95fb6d43868b4488d401c51e9ef4bb4d557abed8b76a41f333a19652c4b2811	3619d2d6e428cc34562e9390bf4acb57

hashlib modülü

BÖLÜM 4. TARTIŞMA VE SONUÇ

Bu çalışmada ana hedef, dünya genelinde popüler bir programlama dili haline gelen Python programlama dilinin şifre bilimi alanındaki uygulamalarına odaklanmak ve bu sahaya giriş yapmak isteyen kişilere temel bir kaynak hazırlamak olmuştur. Bu bağlamda öncelikle, Python programlama dilinin kriptoloji için bir araç olarak kullanılmasına yönelik çeşitli yöntemler incelenmiş ve gerekli temel programlama bilgileri ele alınmıştır. Python'ın kolay ve anlaşılır bir söz dizimine sahip olması, kullanıcıları projelerinde çeşitli amaçlara hizmet etmesi için Python'ı kullanmaya teşvik etmektedir. Özel olarak, kriptoloji alanında halihazırda Python için geliştirilmiş birçok modül, paket ve kütüphane bulunmaktadır ve gün geçtikçe hem bu araçlar geliştirilmekte hem de bu araçların yanına ek olarak yeni araçlar eklenmeye devam etmektedir. Bu nedenle, kriptoloji projelerinde Python programlama dilini kullanmak birçok avantajı beraberinde getirmektedir. Tez kapsamında ele alınan konular, yeni bir kriptoloji algoritmasının Python programlama diliyle geliştirilebilmesi için de gereken temel bilgileri sağlamaktadır. Söz konusu, bu tezin hazırlanış sürecinde Fibonacci Polinomları ile yeni bir kriptoloji algoritması geliştirilmeye başlanmıştır. Bu algoritma, Python kodları ile yakın tarihte bir dergiye gönderilmek üzere yayına hazırlanmaktadır. Geliştirilen yeni şifreleme yönteminin literatüre önemli bir katkı sağlayacağına ve yeni çalışmalarla referans olacağına inanılmaktadır.

KAYNAKLAR

- Ascii Codes: Code Page 857 (Turkish Language). <https://www.ascii-codes.com/cp857.html> (Erişim Tarihi: 19.05.2023).
- Brookshear, J.G., Brylow, D., “Bilgisayar Bilimine Giriş”, 12. Basımdan Çeviri, Nobel Akademik Yayıncılık, Ankara, 2018.
- Cryptography Documentation. <https://cryptography.io/en/latest> (Erişim Tarihi: 31.05.2023).
- Çivi Bilir, G., “Caesar’ın Anahtarı: Geçmişten Günümüze Klasik Şifreleme Yöntemleri”, İTÜ Yayınevi, Baskıda.
- FIPS Publication 197, NIST, 2001. <https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf> (Erişim Tarihi: 19.05.2023).
- Lutz, M., “Programming Python”, 1st Edition, O'Reilly & Associates, Inc., United States of America, 1996.
- NumPy Documentation. <https://numpy.org/doc/stable/index.html> (Erişim Tarihi: 28.05.2023).
- PyCryptodome Documentation. <https://www.pycryptodome.org> (Erişim Tarihi: 28.05.2023).
- Python Developer's Guide. <https://devguide.python.org/versions> (Erişim Tarihi: 21.04.2023).
- Python Documentation. <https://docs.python.org/3> (Erişim Tarihi: 31.05.2023).
- Samancıoğlu, A., “Python Sıfırdan Uzmanlığa Programlama”, 6. Baskı, Unikod Yayıncılık, İstanbul, 2023.
- TIOBE Index. <https://www.tiobe.com/tiobe-index> (Erişim Tarihi: 31.05.2023).
- Trappe, W., Washington, L. C., “Introduction to Cryptography with Coding Theory”, 2nd Edition, Pearson Education, Inc., United States of America, 2006.
- Ural, N., Örenç, Ö., “Uygulamalı Şifreleme ve Şifre Çözme Yöntemleri”, 4. Baskı, Pusula Yayıncılık, İstanbul, 2020.

Teşekkür Ederim ☺