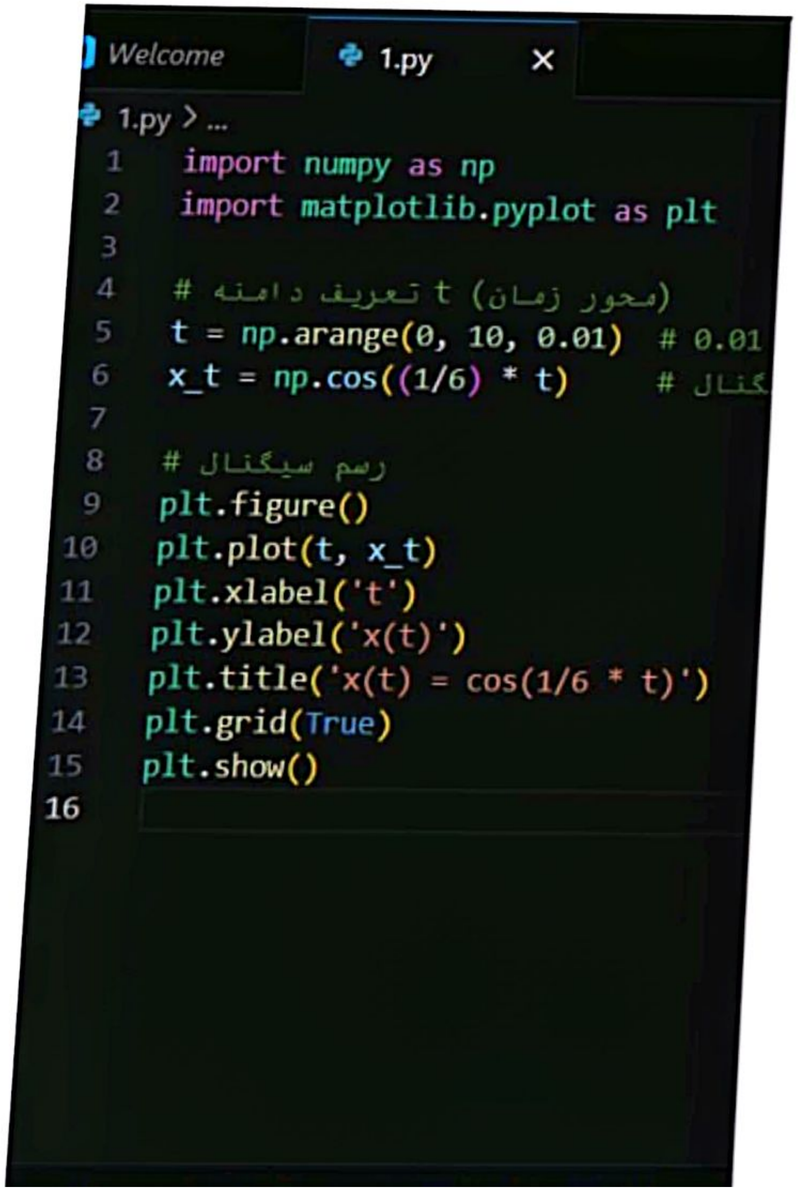
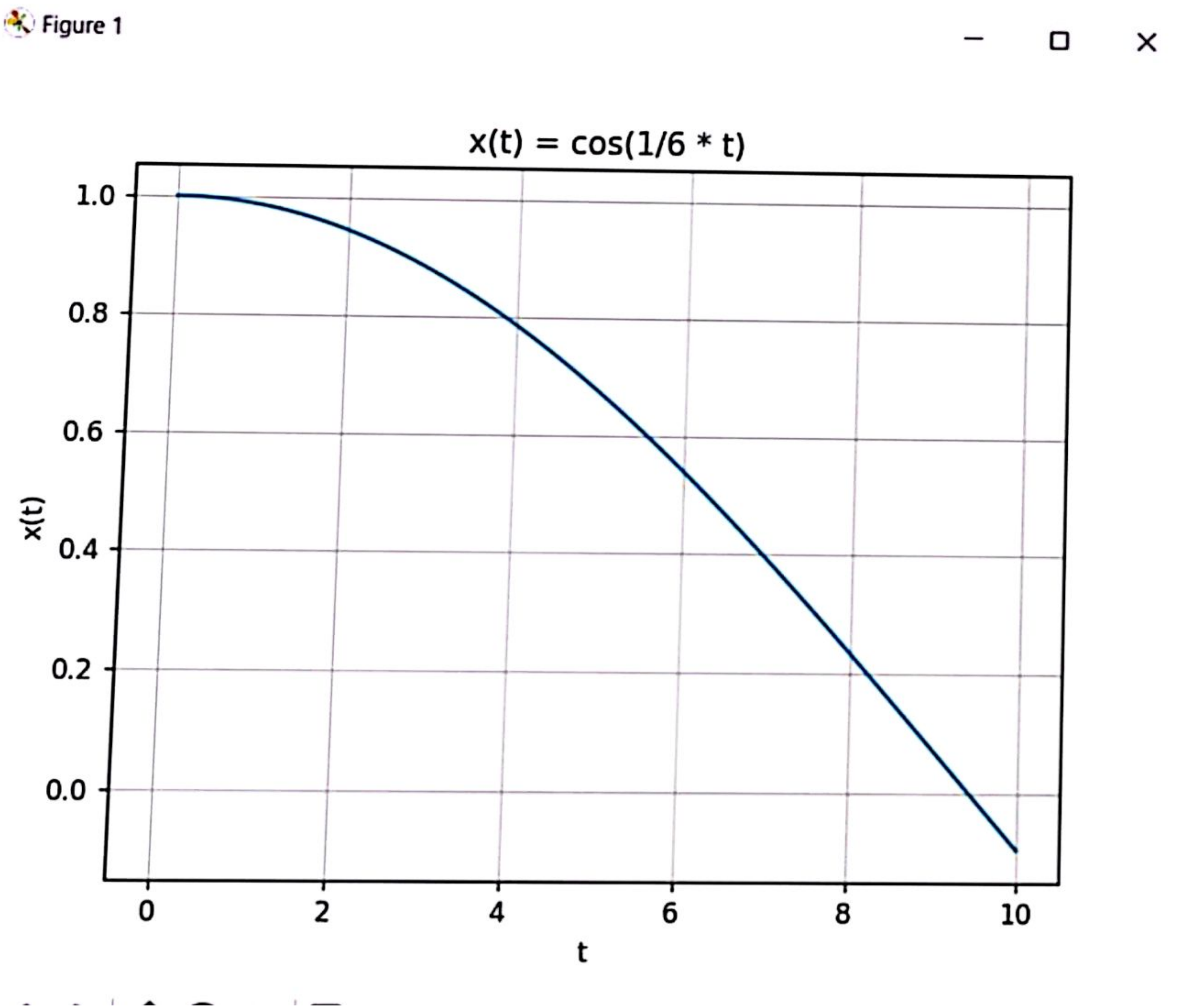
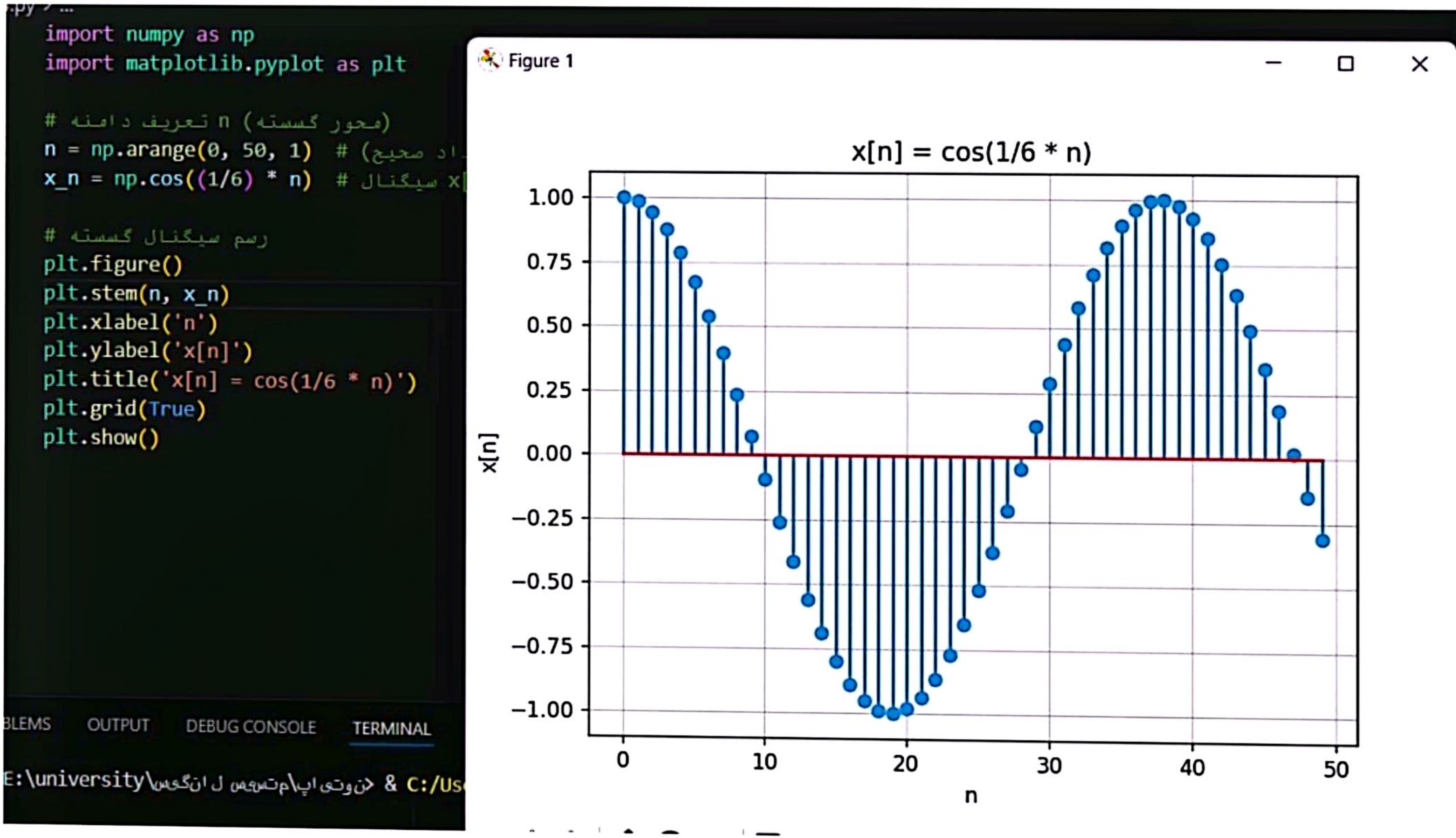


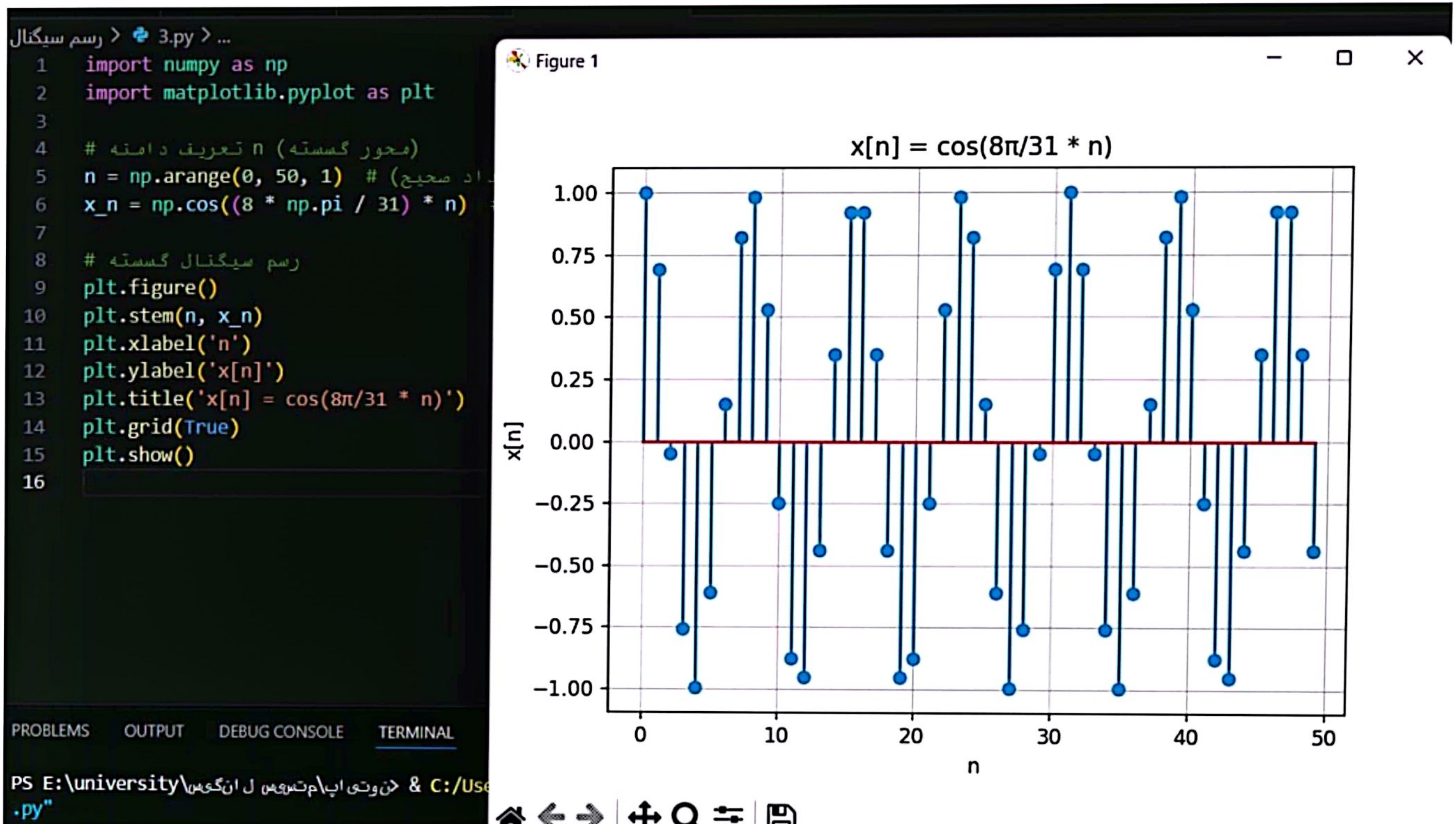
```
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه t (محور زمان)
t = np.arange(0, 10, 0.01)  # 0.01
x_t = np.cos((1/6) * t)      # گنال

# رسم سیگنال
plt.figure()
plt.plot(t, x_t)
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title('x(t) = cos(1/6 * t)')
plt.grid(True)
plt.show()
```





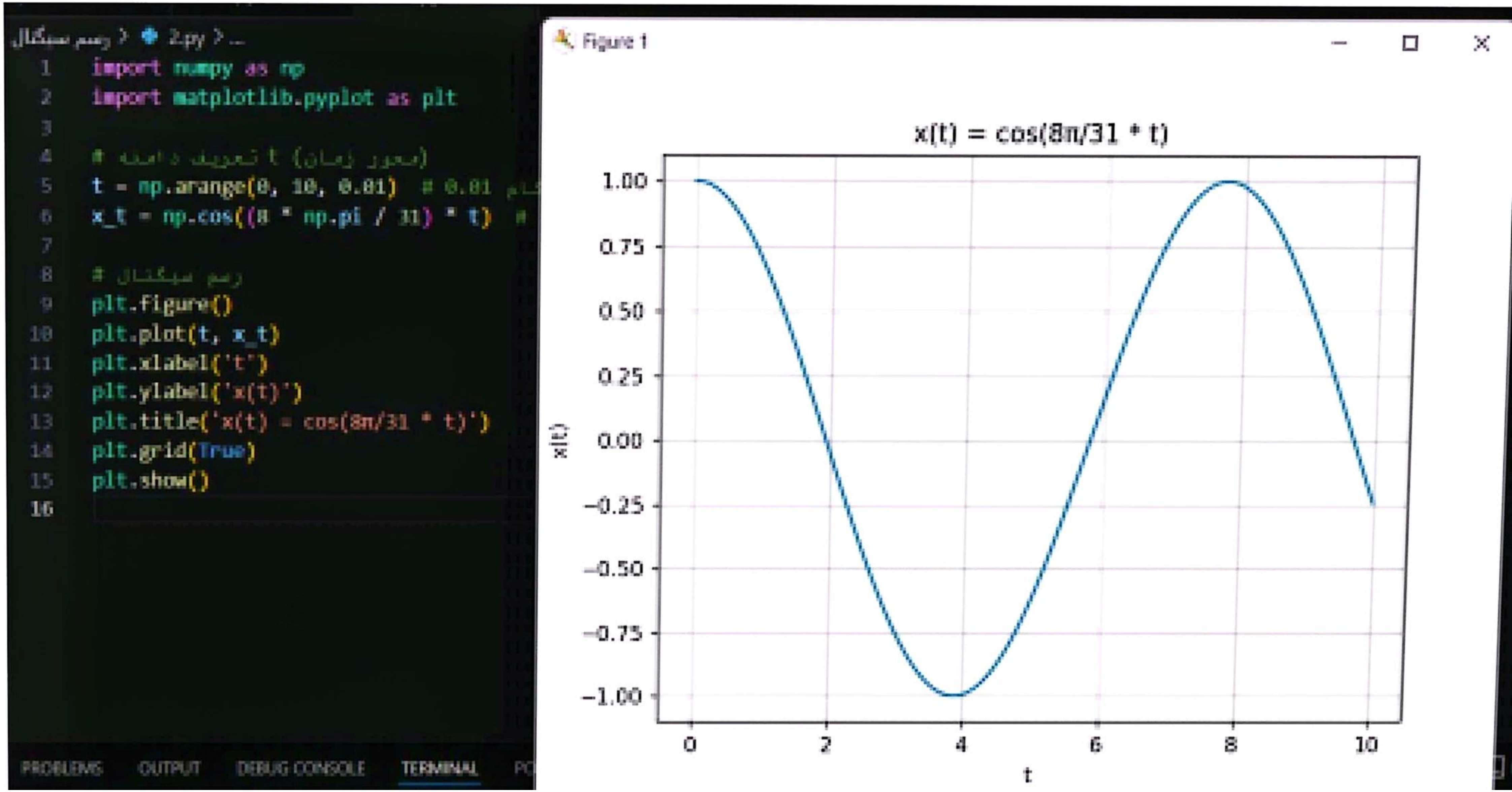
```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(0, 50, 1)  # (صحیح اد
x_n = np.cos((1/6) * n)  # سیگنال x[

# رسم سیگنال گسسته
plt.figure()
plt.stem(n, x_n)
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('x[n] = cos(1/6 * n)')
plt.grid(True)
plt.show()
```

```
E:\university\سیگنال سیستم\پایتون> & C:/Us
```

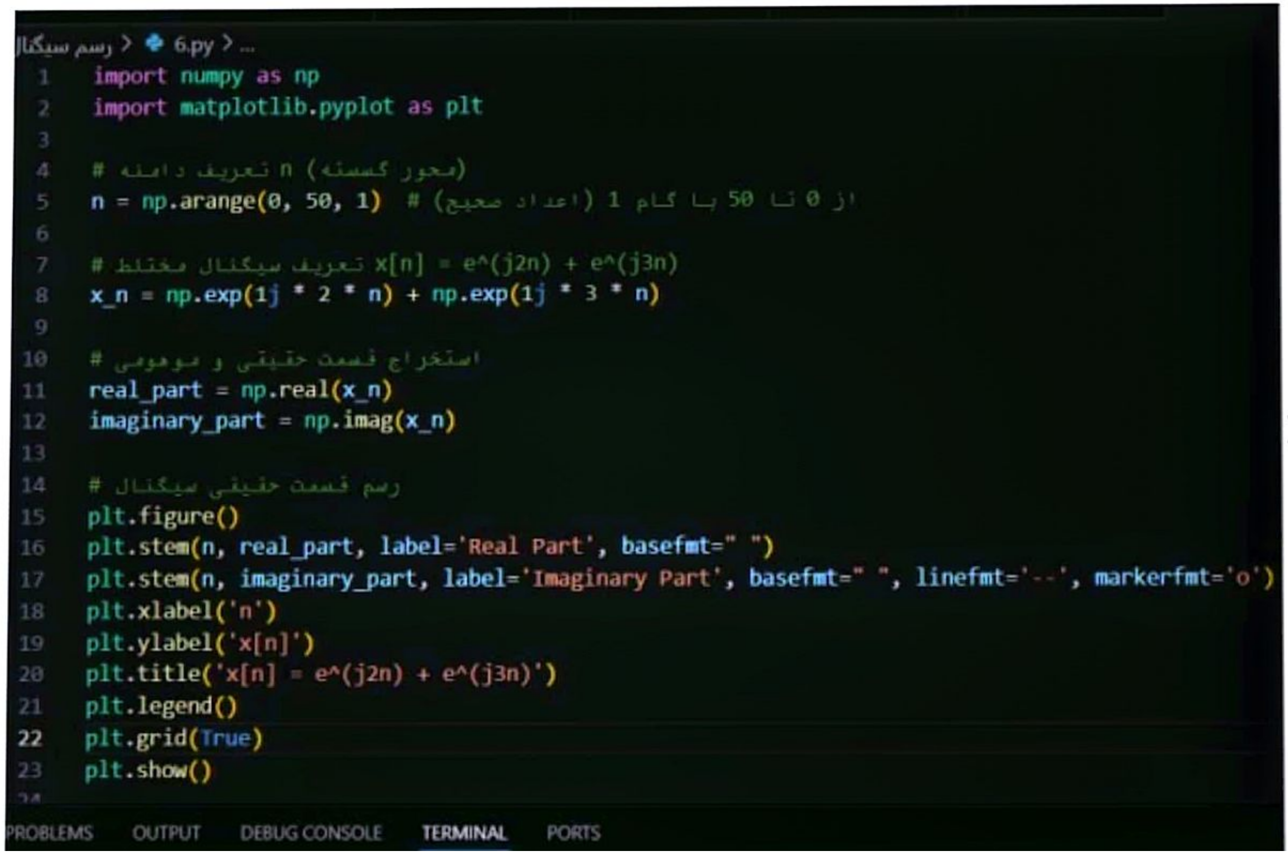


```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(0, 50, 1)  # (صحیح اد
x_n = np.cos((8 * np.pi / 31) * n)

# رسم سیگنال گسسته
plt.figure()
plt.stem(n, x_n)
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('x[n] = cos(8π/31 * n)')
plt.grid(True)
plt.show()

```

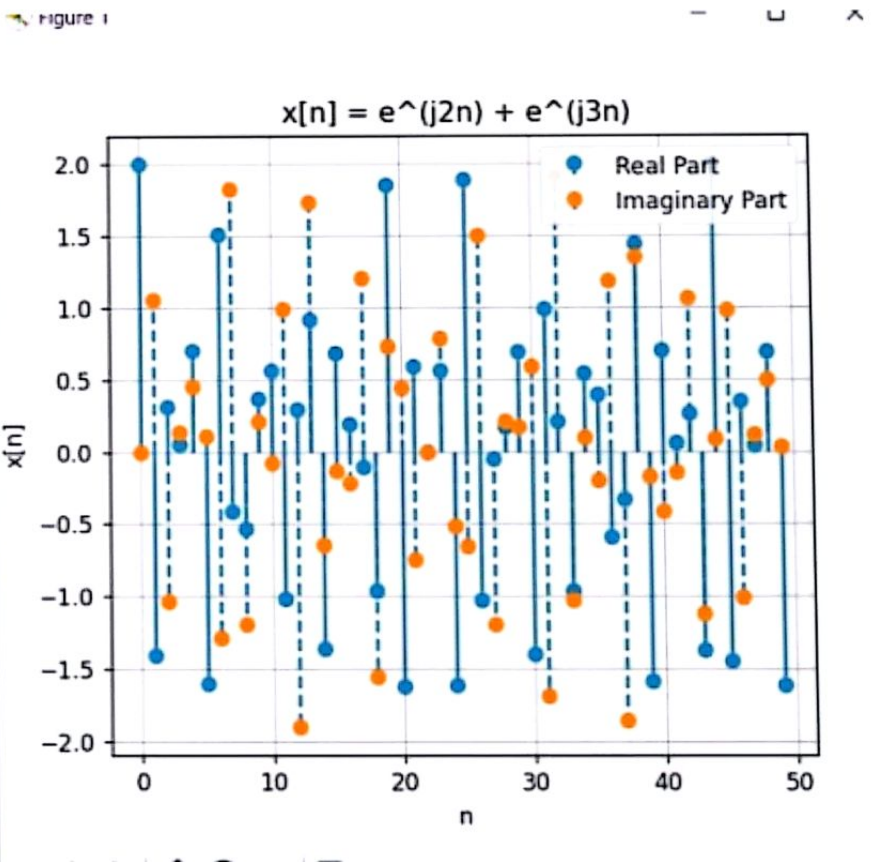
```
PS E:\university\سیگنال و سیستم\پایتون> & C:/Use
.py"
```

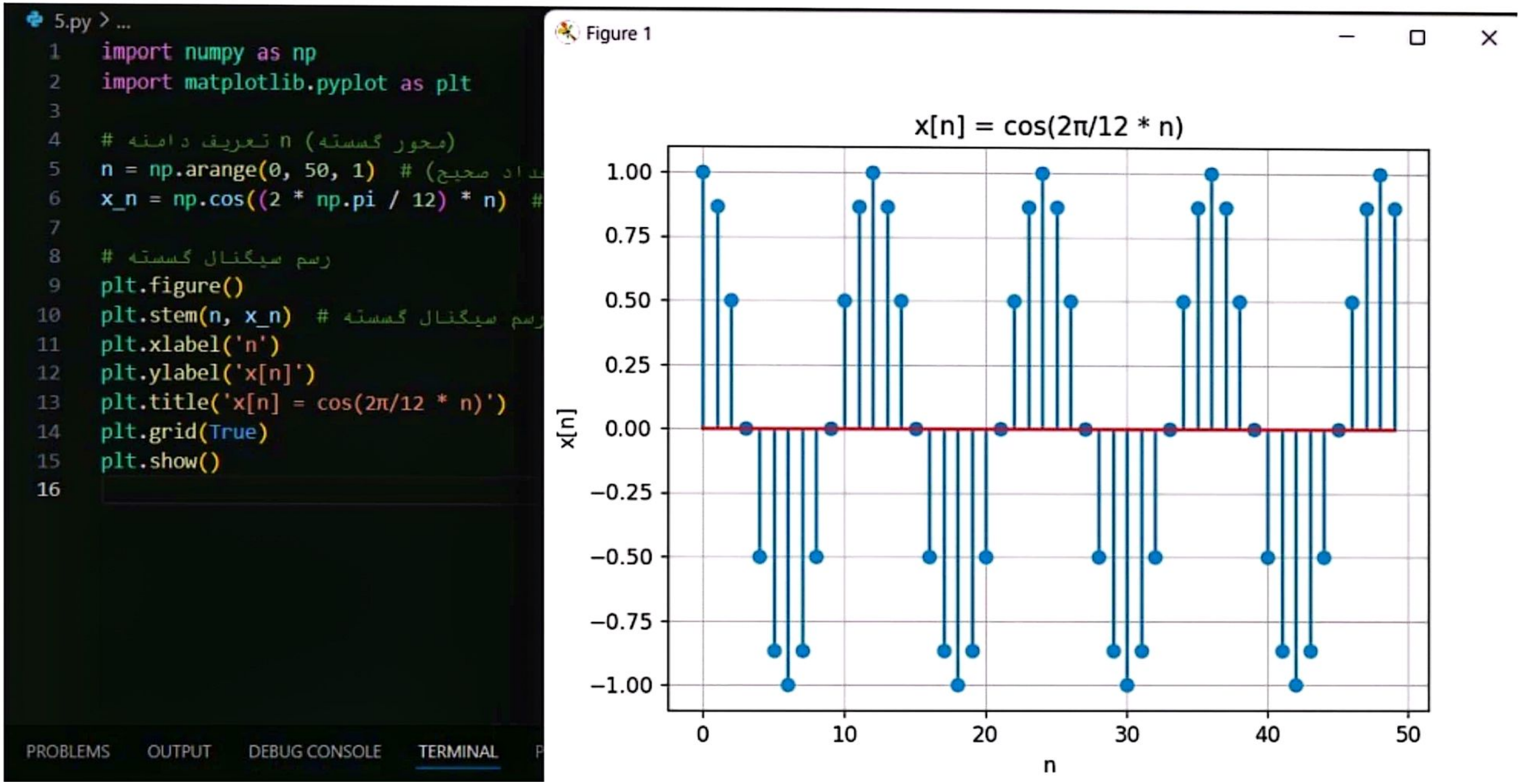


```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه t (محور زمان)
t = np.arange(0, 10, 0.01)  # 0.01
x_t = np.cos((8 * np.pi / 31) * t)  #

# رسم سیگنال
plt.figure()
plt.plot(t, x_t)
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title('x(t) = cos(8π/31 * t)')
plt.grid(True)
plt.show()

```

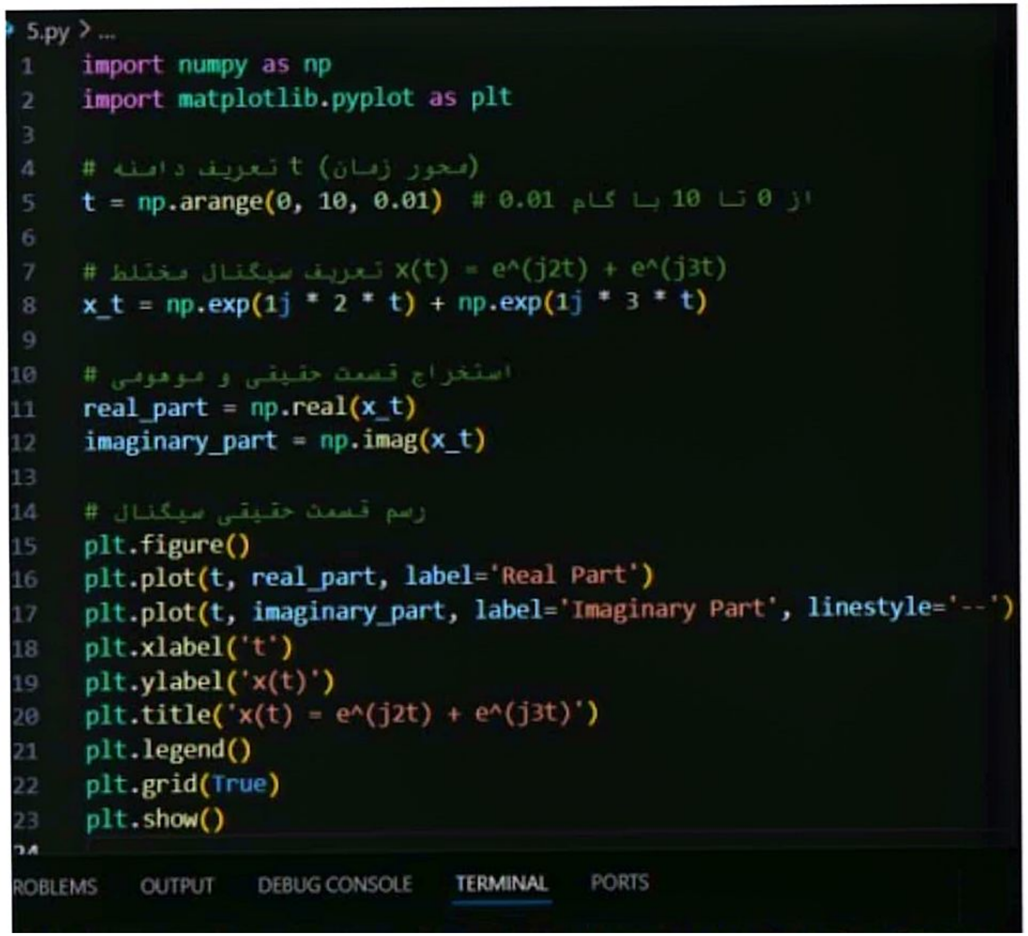


```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(0, 50, 1)  # از 0 تا 50 با گام 1 (اعداد صحیح)

# تعریف سیگنال مختلط x[n] = e^(j2n) + e^(j3n)
x_n = np.exp(1j * 2 * n) + np.exp(1j * 3 * n)

# استخراج قسمت حقیقی و موهومی
real_part = np.real(x_n)
imaginary_part = np.imag(x_n)

# رسم قسمت حقیقی سیگنال
plt.figure()
plt.stem(n, real_part, label='Real Part', basefmt=" ")
plt.stem(n, imaginary_part, label='Imaginary Part', basefmt=" ", linefmt='--', markerfmt='o')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('x[n] = e^(j2n) + e^(j3n)')
plt.legend()
plt.grid(True)
plt.show()
```

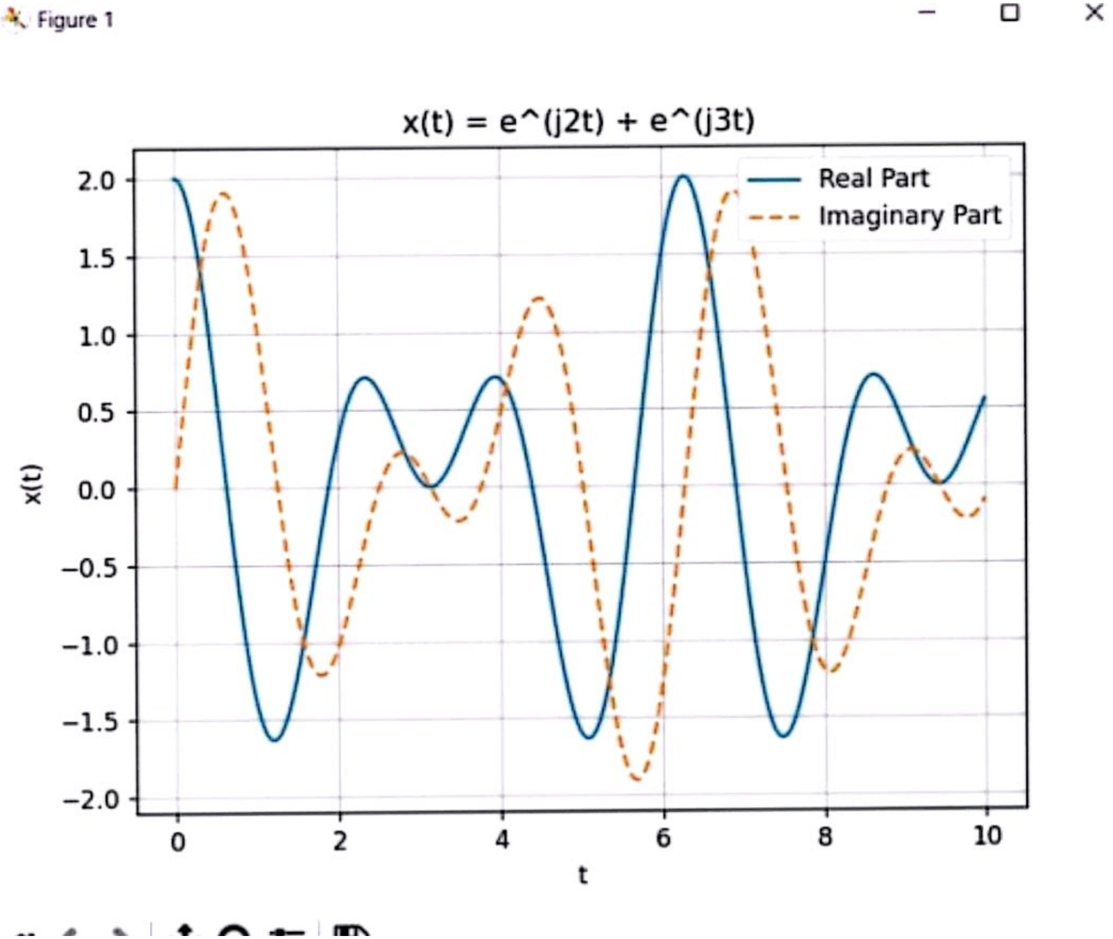


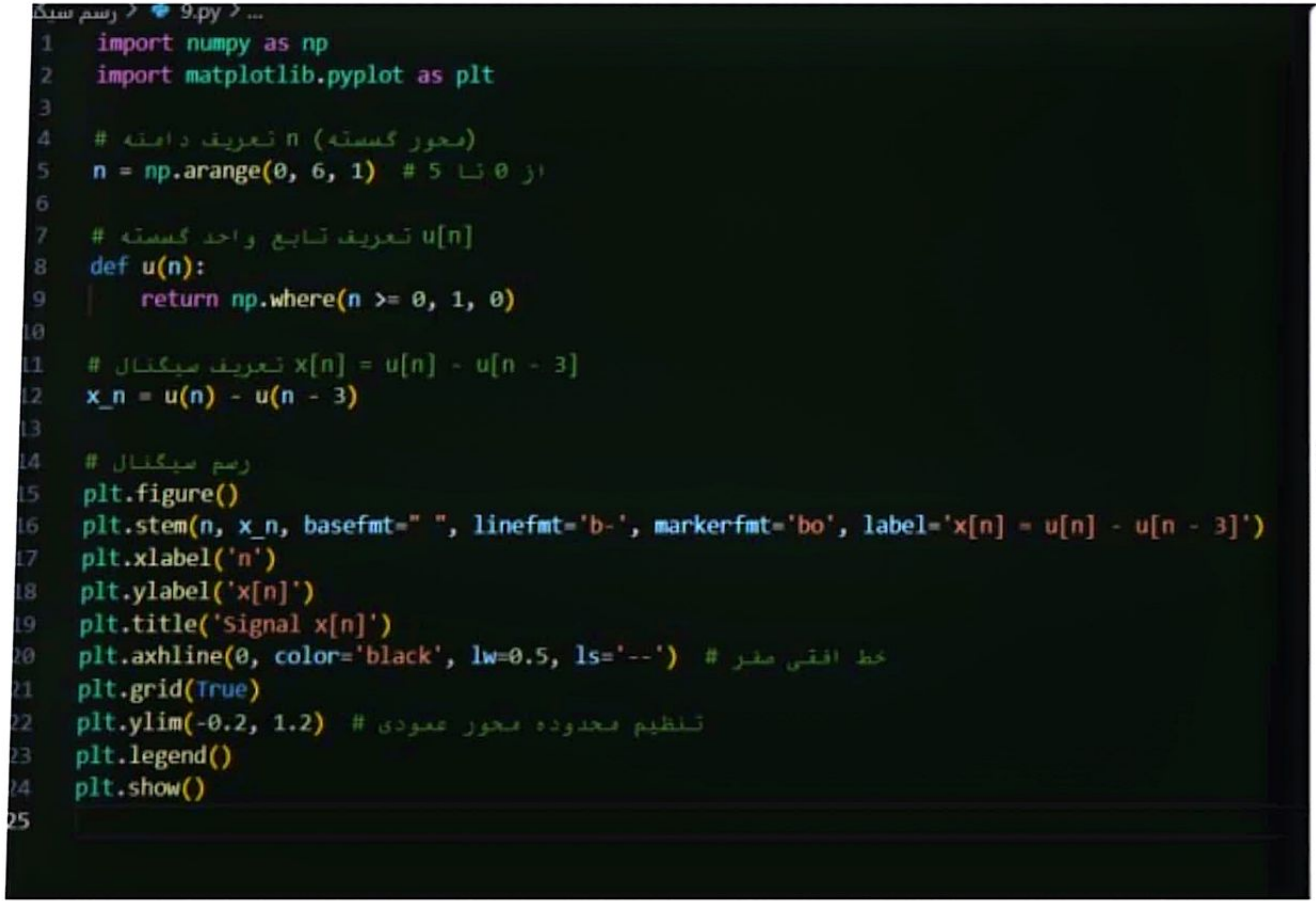


```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(0, 50, 1)  # داد صحیح)
x_n = np.cos((2 * np.pi / 12) * n)  #

# رسم سیگنال گسسته
plt.figure()
plt.stem(n, x_n)  # رسم سیگنال گسسته
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('x[n] = cos(2π/12 * n)')
plt.grid(True)
plt.show()

```

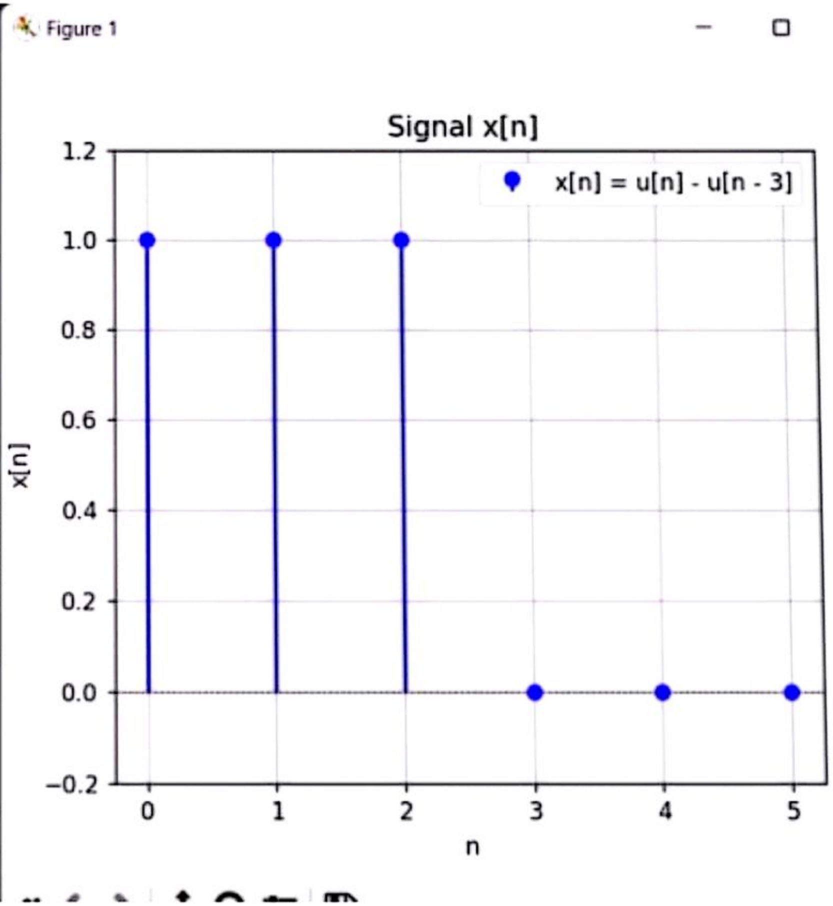


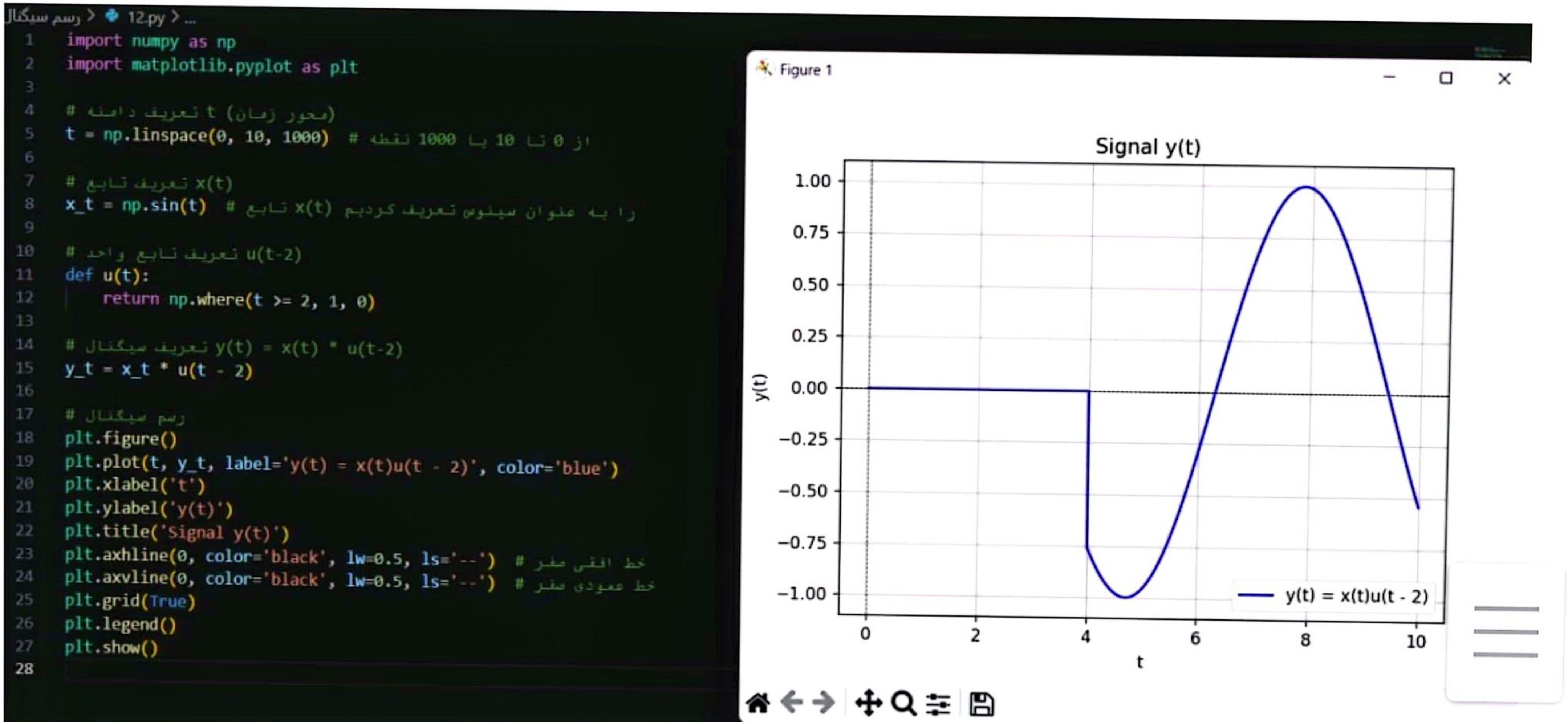
```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه t (محور زمان)
t = np.arange(0, 10, 0.01)  # از 0 تا 10 با گام 0.01

# تعریف سیگنال مختلط x(t) = e^(j2t) + e^(j3t)
x_t = np.exp(1j * 2 * t) + np.exp(1j * 3 * t)

# استخراج قسمت حقیقی و موهومی
real_part = np.real(x_t)
imaginary_part = np.imag(x_t)

# رسم قسمت حقیقی سیگنال
plt.figure()
plt.plot(t, real_part, label='Real Part')
plt.plot(t, imaginary_part, label='Imaginary Part', linestyle='--')
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title('x(t) = e^(j2t) + e^(j3t)')
plt.legend()
plt.grid(True)
plt.show()
```



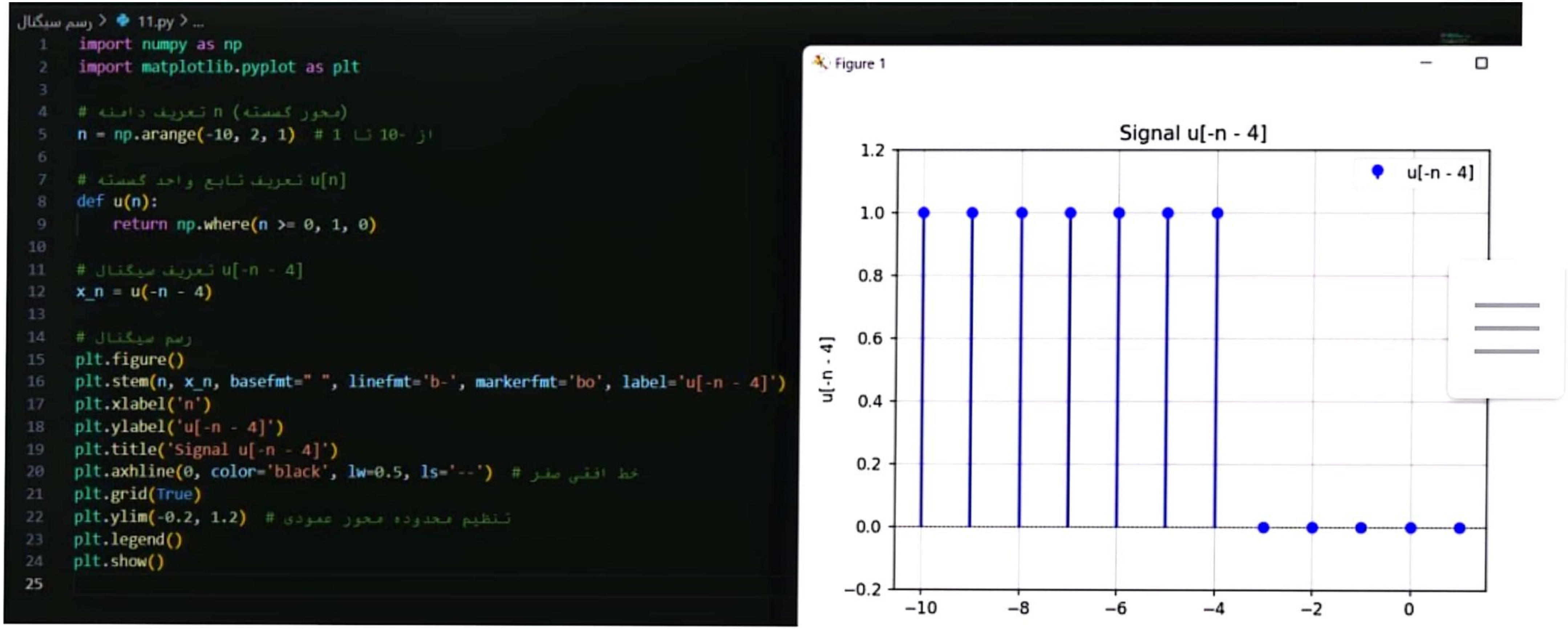


```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(0, 6, 1)  # از 0 تا 5

# تعریف تابع واحد گسسته u[n]
def u(n):
    return np.where(n >= 0, 1, 0)

# تعریف سیگنال x[n] = u[n] - u[n - 3]
x_n = u(n) - u(n - 3)

# رسم سیگنال
plt.figure()
plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='x[n] = u[n] - u[n - 3]')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('Signal x[n]')
plt.axhline(0, color='black', lw=0.5, ls='--')  # خط افقی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2)  # تنظیم محدوده محور عمودی
plt.legend()
plt.show()
```

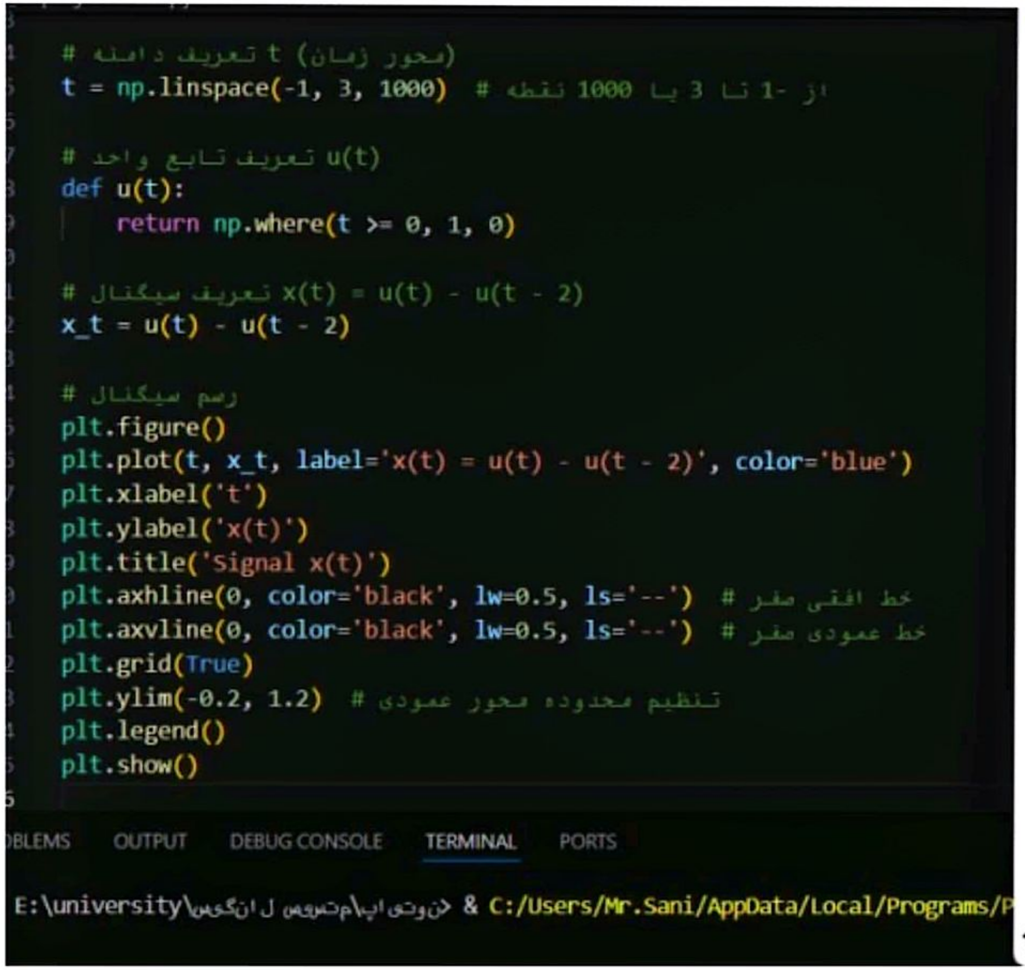




```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه t (محور زمان)
t = np.linspace(0, 10, 1000)  # از 0 تا 10 با 1000 نقطه

# تعریف تابع x(t)
x_t = np.sin(t)  # تابع x(t) را به عنوان سینوس تعریف کردیم

# تعریف تابع واحد u(t-2)
def u(t):
    return np.where(t >= 2, 1, 0)

# تعریف سیگنال y(t) = x(t) * u(t-2)
y_t = x_t * u(t - 2)

# رسم سیگنال
plt.figure()
plt.plot(t, y_t, label='y(t) = x(t)u(t - 2)', color='blue')
plt.xlabel('t')
plt.ylabel('y(t)')
plt.title('Signal y(t)')
plt.axhline(0, color='black', lw=0.5, ls='--')  # خط افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--')  # خط عمودی صفر
plt.grid(True)
plt.legend()
plt.show()

```

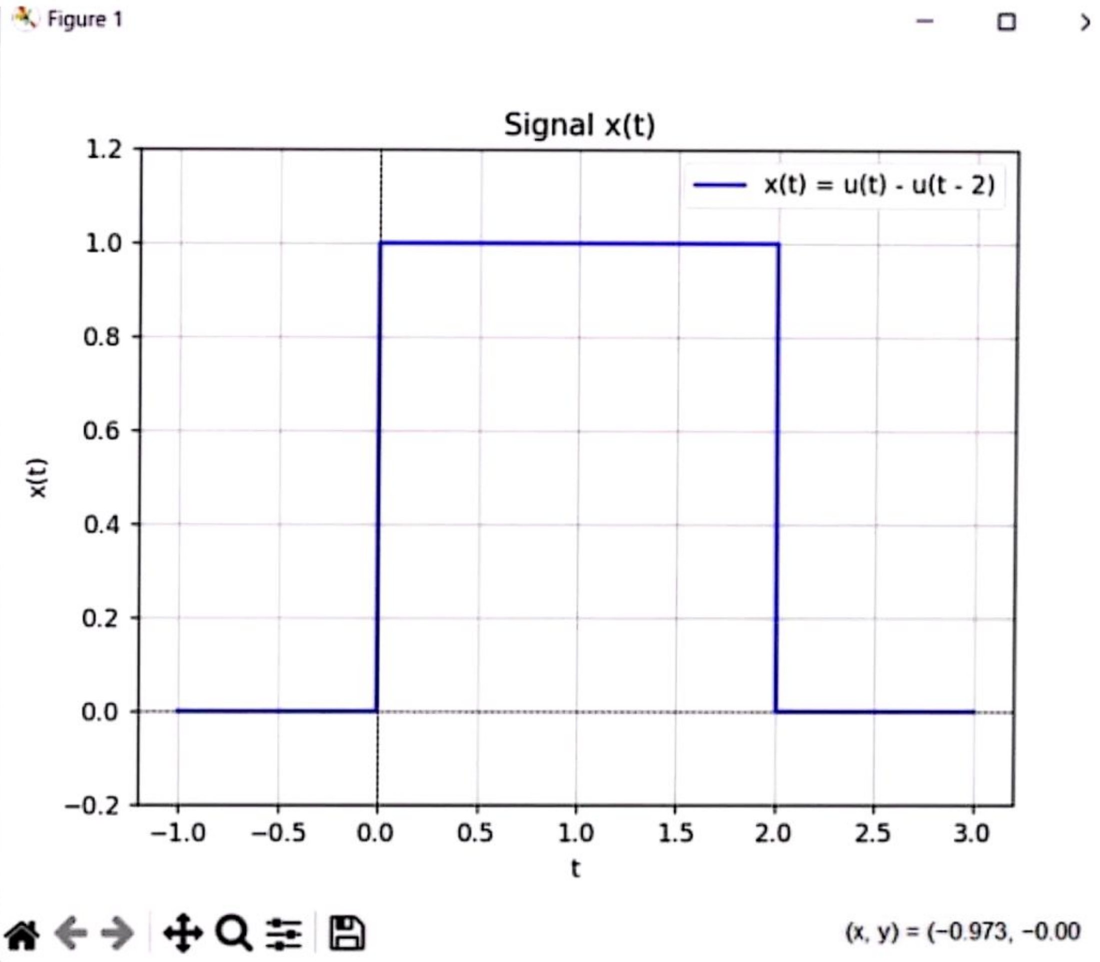


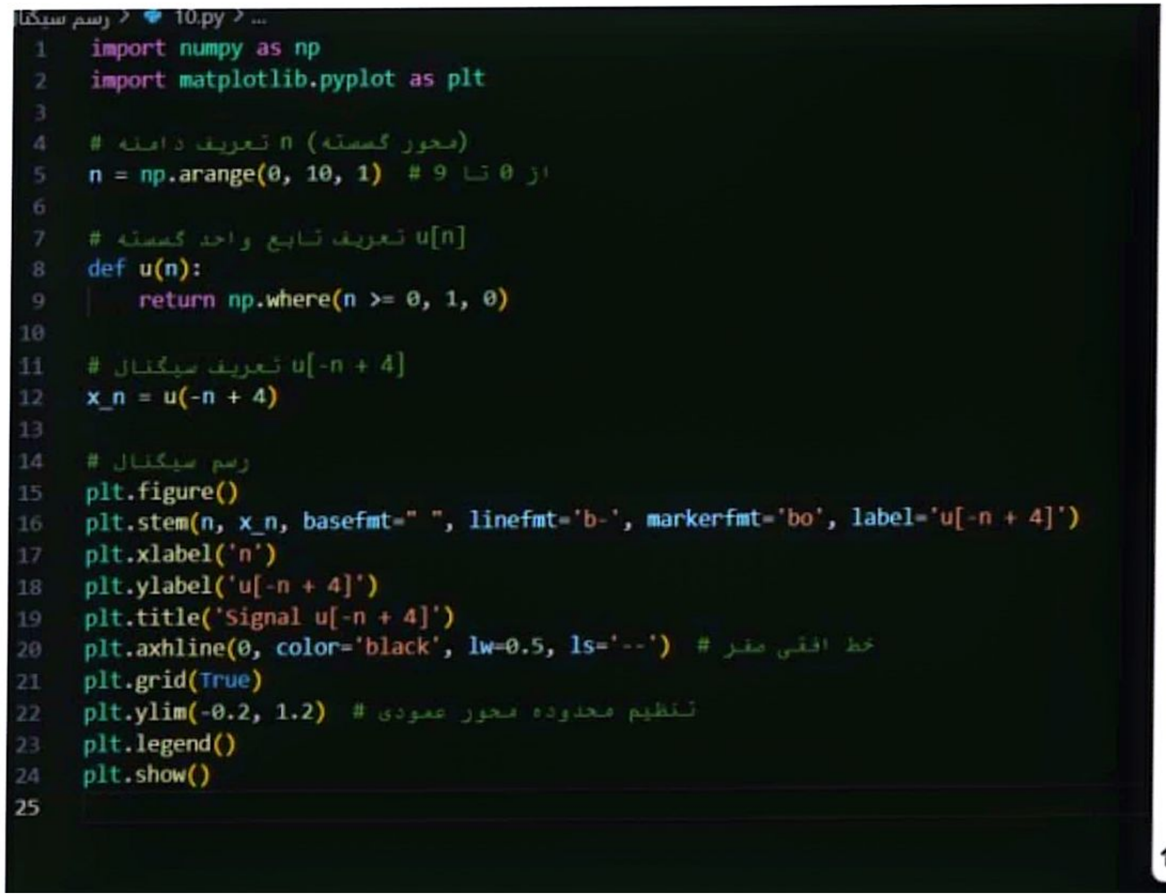
```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(-10, 2, 1)  # از -10 تا 1

# تعریف تابع واحد گسسته u[n]
def u(n):
    return np.where(n >= 0, 1, 0)

# تعریف سیگنال u[-n - 4]
x_n = u(-n - 4)

# رسم سیگنال
plt.figure()
plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='u[-n - 4]')
plt.xlabel('n')
plt.ylabel('u[-n - 4]')
plt.title('Signal u[-n - 4]')
plt.axhline(0, color='black', lw=0.5, ls='--')  # خط افقی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2)  # تنظیم محدوده محور عمودی
plt.legend()
plt.show()

```



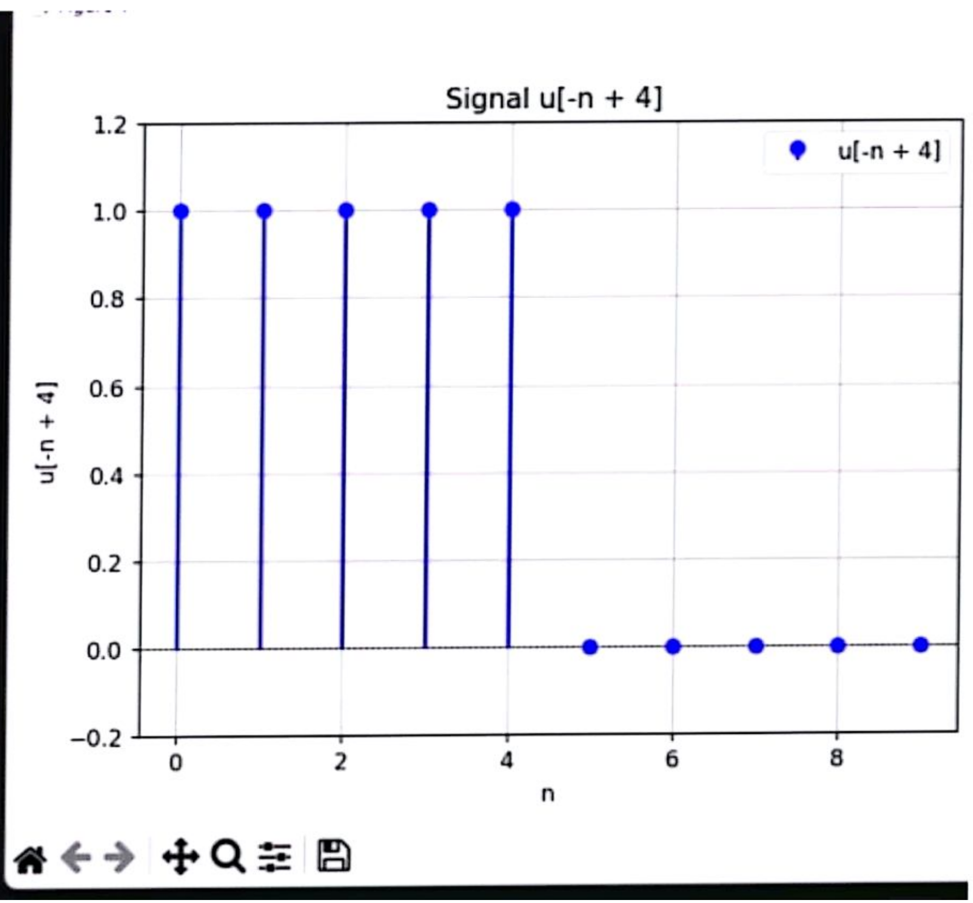
```
# تعریف دامنه t (محور زمان)
t = np.linspace(-1, 3, 1000)  # از -1 تا 3 با 1000 نقطه

# تعریف تابع واحد u(t)
def u(t):
    return np.where(t >= 0, 1, 0)

# تعریف سیگنال x(t) = u(t) - u(t - 2)
x_t = u(t) - u(t - 2)

# رسم سیگنال
plt.figure()
plt.plot(t, x_t, label='x(t) = u(t) - u(t - 2)', color='blue')
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title('Signal x(t)')
plt.axhline(0, color='black', lw=0.5, ls='--')  # خط افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--')  # خط عمودی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2)  # تنظیم محدوده محور عمودی
plt.legend()
plt.show()
```

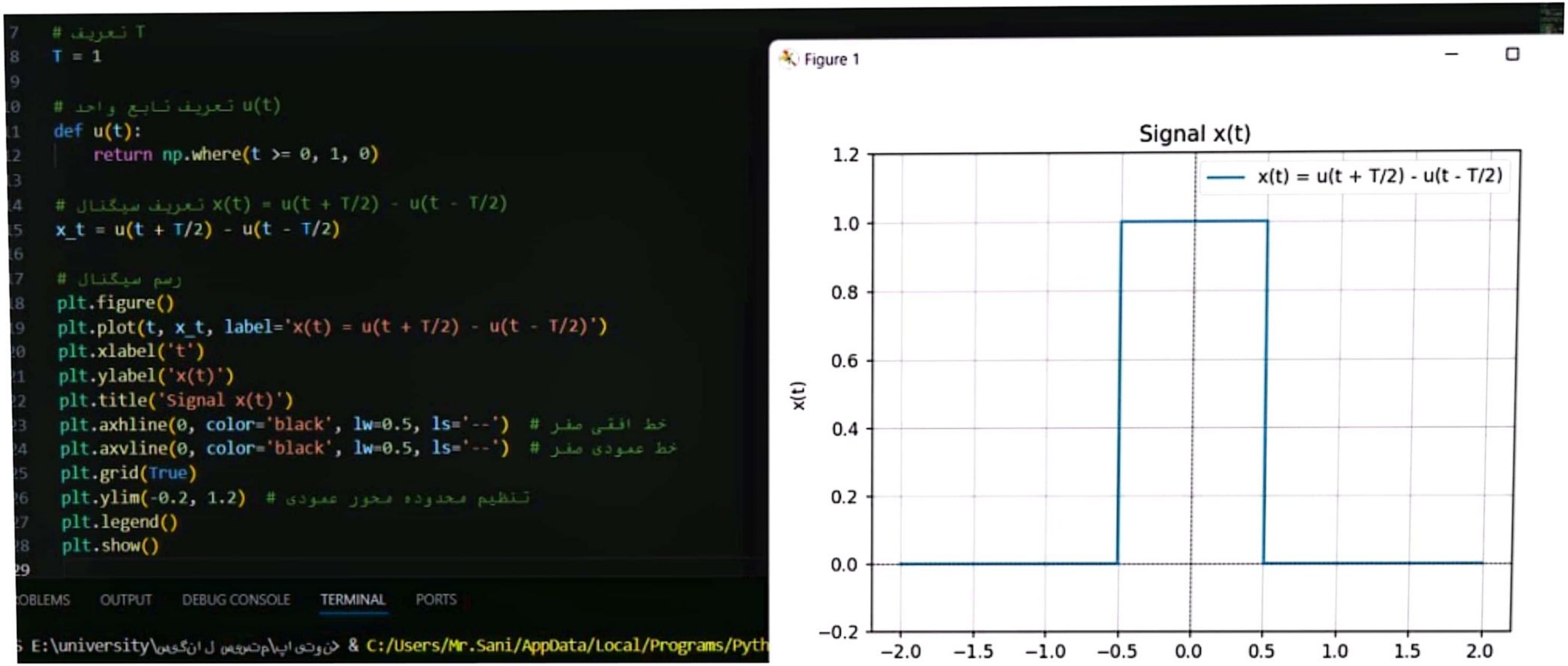
```
E:\university\سیگنال سیستم\پایتون> & C:/Users/Mr.Sani/AppData/Local/Programs/P
```





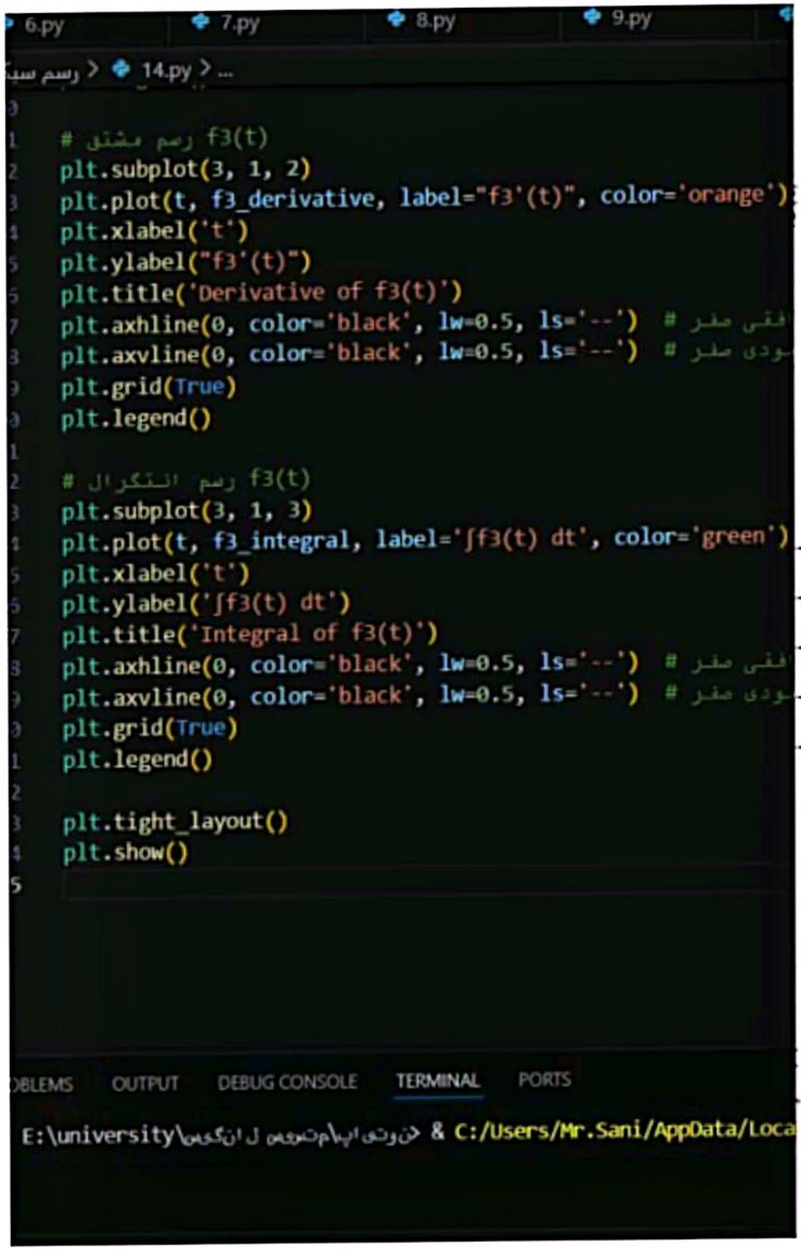
```python
import numpy as np
import matplotlib.pyplot as plt

# تعریف دامنه n (محور گسسته)
n = np.arange(0, 10, 1)  # از 0 تا 9

# تعریف تابع واحد گسسته u[n]
def u(n):
    return np.where(n >= 0, 1, 0)

# تعریف سیگنال u[-n + 4]
x_n = u(-n + 4)

# رسم سیگنال
plt.figure()
plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='u[-n + 4]')
plt.xlabel('n')
plt.ylabel('u[-n + 4]')
plt.title('Signal u[-n + 4]')
plt.axhline(0, color='black', lw=0.5, ls='--')  # خط افقی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2)  # تنظیم محدوده محور عمودی
plt.legend()
plt.show()
```





```
# تعریف T
T = 1

# تعریف تابع واحد u(t)
def u(t):
    return np.where(t >= 0, 1, 0)

# تعریف سیگنال x(t) = u(t + T/2) - u(t - T/2)
x_t = u(t + T/2) - u(t - T/2)

# رسم سیگنال
plt.figure()
plt.plot(t, x_t, label='x(t) = u(t + T/2) - u(t - T/2)')
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title('Signal x(t)')
plt.axhline(0, color='black', lw=0.5, ls='--')  # خط افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--')  # خط عمودی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2)  # تنظیم محدوده محور عمودی
plt.legend()
plt.show()
```

```
S E:\university\سیگنال سیستم\پایتون> & C:/Users/Mr.Sani/AppData/Local/Programs/Pyth
```



```python
# رسم مشتق f3(t)
plt.subplot(3, 1, 2)
plt.plot(t, f3_derivative, label="f3'(t)", color='orange')
plt.xlabel('t')
plt.ylabel("f3'(t)")
plt.title('Derivative of f3(t)')
plt.axhline(0, color='black', lw=0.5, ls='--')  # افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--')  # ودی صفر
plt.grid(True)
plt.legend()

# رسم انتگرال f3(t)
plt.subplot(3, 1, 3)
plt.plot(t, f3_integral, label='∫f3(t) dt', color='green')
plt.xlabel('t')
plt.ylabel('∫f3(t) dt')
plt.title('Integral of f3(t)')
plt.axhline(0, color='black', lw=0.5, ls='--')  # افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--')  # ودی صفر
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()
```

```
E:\university\سیگنال و سیستم\پایتون> & C:/Users/Mr.Sani/AppData/Loca
```