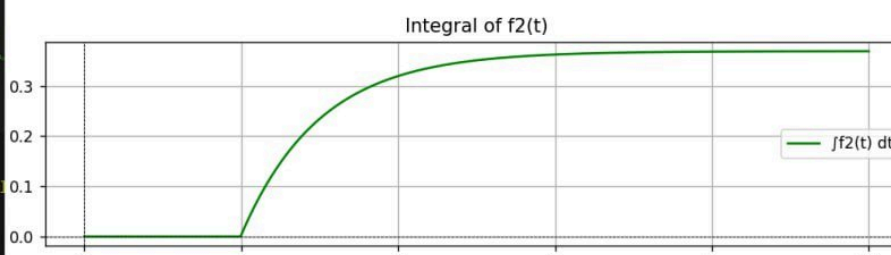
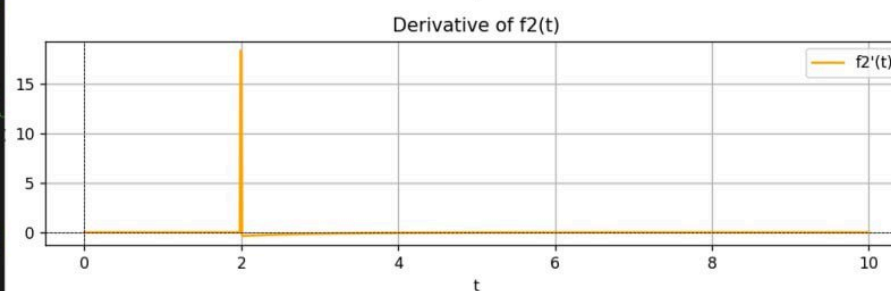
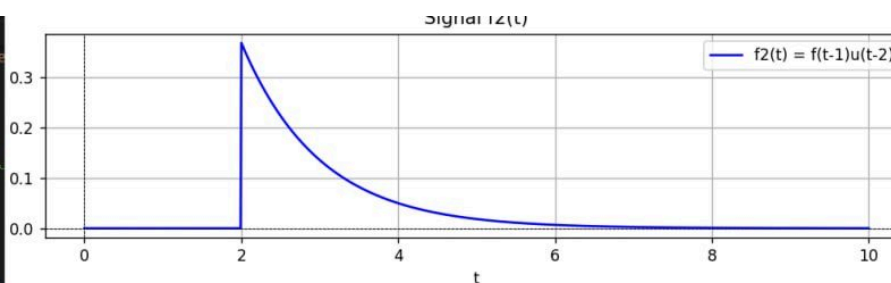


```

> 13.py > ...
32 plt.plot(t, f2_t, label='f2(t) = f(t-1)u(t-2)', color='blue')
33 plt.xlabel('t')
34 plt.ylabel('f2(t)')
35 plt.title('Signal f2(t)')
36 plt.axhline(0, color='black', lw=0.5, ls='--') # افقی محور
37 plt.axvline(0, color='black', lw=0.5, ls='--') # عمودی محور
38 plt.grid(True)
39 plt.legend()
40
41 # رسم مشتق f2(t)
42 plt.subplot(3, 1, 2)
43 plt.plot(t, f2_derivative, label="f2'(t)", color='orange')
44 plt.xlabel('t')
45 plt.ylabel("f2'(t)")
46 plt.title('Derivative of f2(t)')
47 plt.axhline(0, color='black', lw=0.5, ls='--') # افقی محور
48 plt.axvline(0, color='black', lw=0.5, ls='--') # عمودی محور
49 plt.grid(True)
50 plt.legend()
51
52 # رسم انتگرال f2(t)
53 plt.subplot(3, 1, 3)
54 plt.plot(t, f2_integral, label='∫f2(t) dt', color='green')
55 plt.xlabel('t')
56 plt.ylabel('∫f2(t) dt')
57 plt.title('Integral of f2(t)')
58 plt.axhline(0, color='black', lw=0.5, ls='--') # افقی محور
59 plt.axvline(0, color='black', lw=0.5, ls='--') # عمودی محور
60 plt.grid(True)
61 plt.legend()
62
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
S E:\university\انگلیسی لایه های پایانی & C:\Users\Mr.Sani\AppData\Local

```



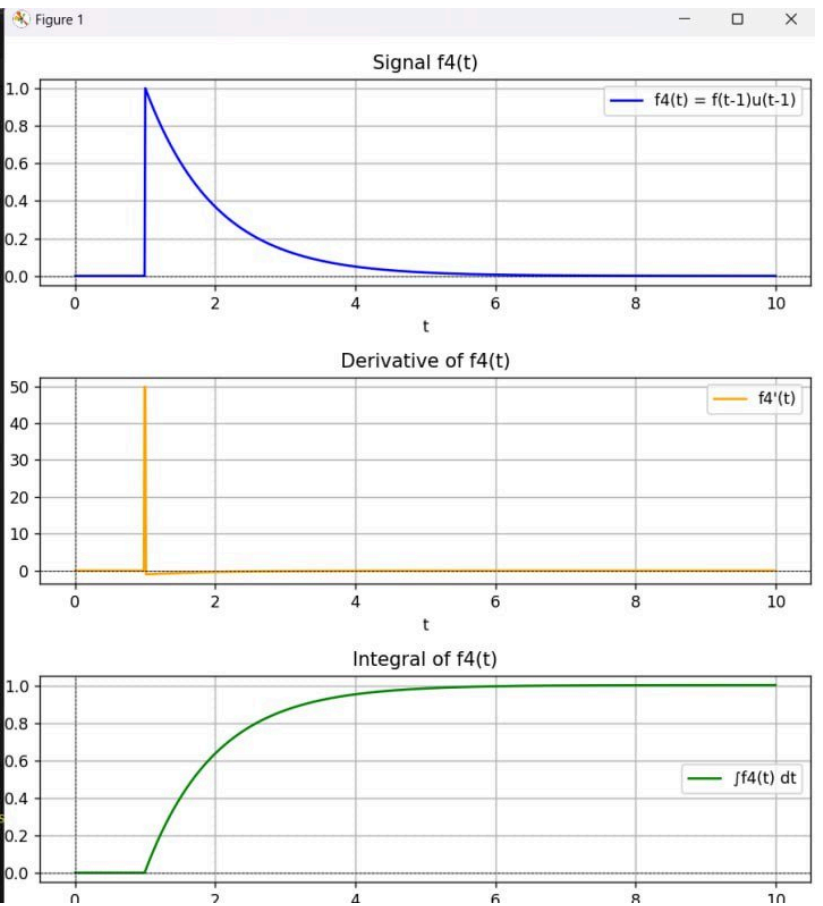
```

6.py 7.py 8.py 9.py 10.py
رسم سیگنال > 15.py > ...
34 plt.ylabel('f4(t)')
35 plt.title('Signal f4(t)')
36 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
37 plt.axvline(0, color='black', lw=0.5, ls='--') # خط عمودی صفر
38 plt.grid(True)
39 plt.legend()
40
41 # رسم مشتق f4(t)
42 plt.subplot(3, 1, 2)
43 plt.plot(t, f4_derivative, label="f4'(t)", color='orange')
44 plt.xlabel('t')
45 plt.ylabel("f4'(t)")
46 plt.title('Derivative of f4(t)')
47 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
48 plt.axvline(0, color='black', lw=0.5, ls='--') # خط عمودی صفر
49 plt.grid(True)
50 plt.legend()
51
52 # رسم انتگرال f4(t)
53 plt.subplot(3, 1, 3)
54 plt.plot(t, f4_integral, label='f4(t) dt', color='green')
55 plt.xlabel('t')
56 plt.ylabel('f4(t) dt')
57 plt.title('Integral of f4(t)')
58 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
59 plt.axvline(0, color='black', lw=0.5, ls='--') # خط عمودی صفر
60 plt.grid(True)
61 plt.legend()
62
63 plt.tight_layout()
64 plt.show()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

S E:\university\انگلیسی ل انگریسی & C:/Users/Mr.Sani/AppData/Local/Programs



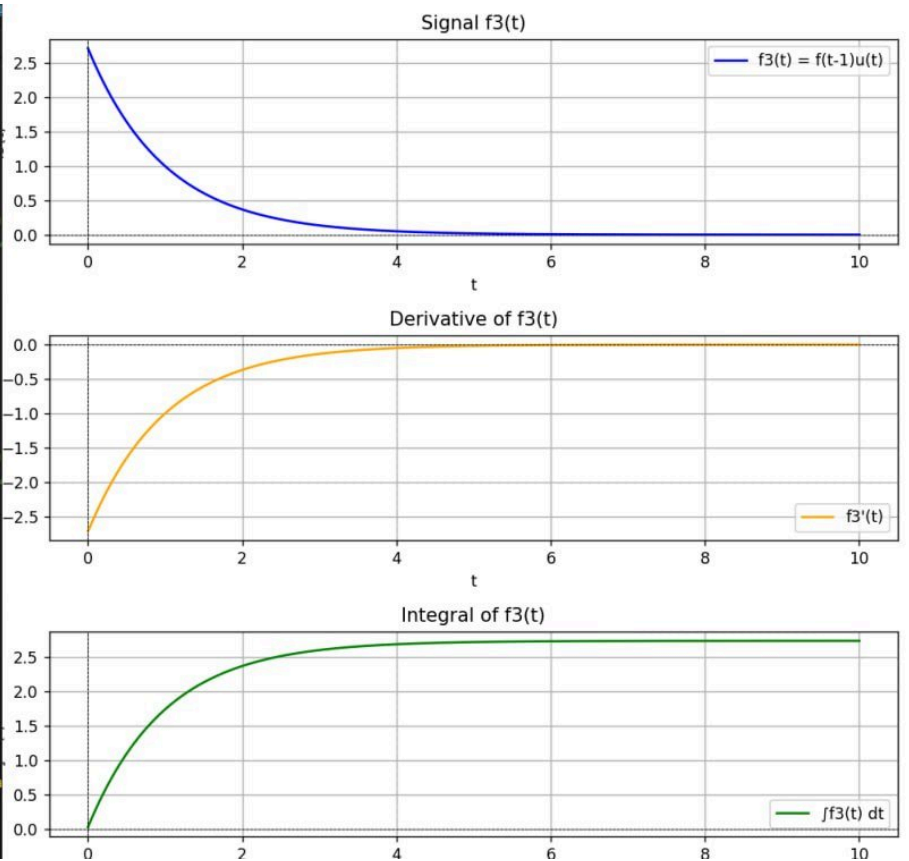
```

6.py 7.py 8.py 9.py
> 14.py > ...
# 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='--') # خطی صفر
plt.axvline(0, color='black', lw=0.5, ls='--') # خطی 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='--') # خطی صفر
plt.axvline(0, color='black', lw=0.5, ls='--') # خطی صفر
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()

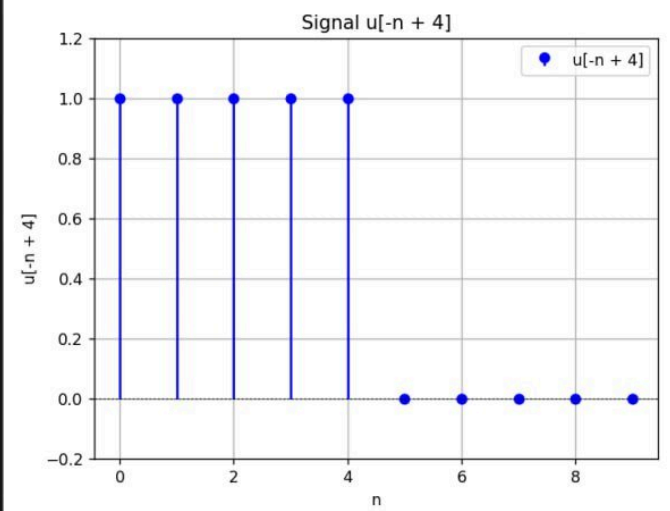
```



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور گسسته n تعریف دامنه
5 n = np.arange(0, 10, 1) # 0 تا 9
6
7 # تعریف تابع واحد گسسته u[n]
8 def u(n):
9     return np.where(n >= 0, 1, 0)
10
11 # تعریف سیگنال u[-n + 4]
12 x_n = u(-n + 4)
13
14 # رسم سیگنال
15 plt.figure()
16 plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='u[-n + 4]')
17 plt.xlabel('n')
18 plt.ylabel('u[-n + 4]')
19 plt.title('Signal u[-n + 4]')
20 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی منفر
21 plt.grid(True)
22 plt.ylim(-0.2, 1.2) # تنظیم محدوده محور عمودی
23 plt.legend()
24 plt.show()
25

```



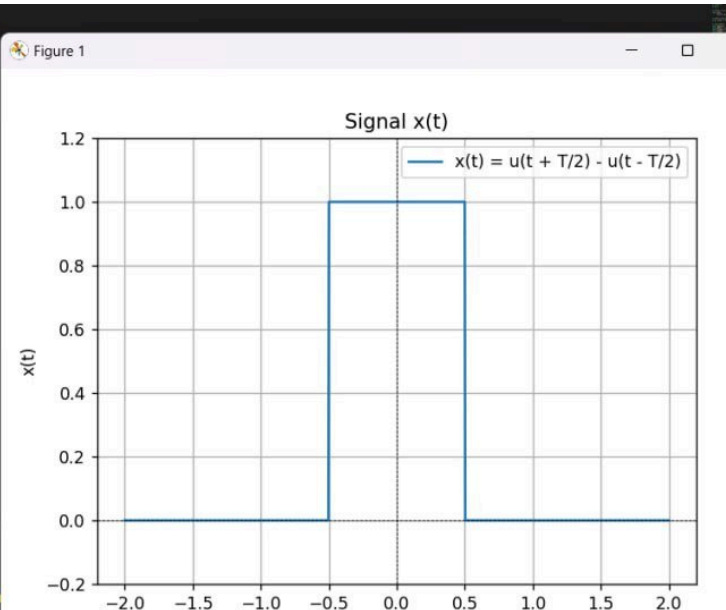
```

7  # تعريف T
8  T = 1
9
10 # تعريف تابع واحد u(t)
11 def u(t):
12     return np.where(t >= 0, 1, 0)
13
14 # تعريف سيگنال x(t) = u(t + T/2) - u(t - T/2)
15 x_t = u(t + T/2) - u(t - T/2)
16
17 # رسم سيگنال
18 plt.figure()
19 plt.plot(t, x_t, label='x(t) = u(t + T/2) - u(t - T/2)')
20 plt.xlabel('t')
21 plt.ylabel('x(t)')
22 plt.title('Signal x(t)')
23 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقي صفر
24 plt.axvline(0, color='black', lw=0.5, ls='--') # خط عمودي صفر
25 plt.grid(True)
26 plt.ylim(-0.2, 1.2) # تنظيم محدوده محور عمودي
27 plt.legend()
28 plt.show()
29

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

\$ E:\university\انگلیس ل\متن و تصاویر & C:/Users/Mr.Sani/AppData/Local/Programs/Python



```

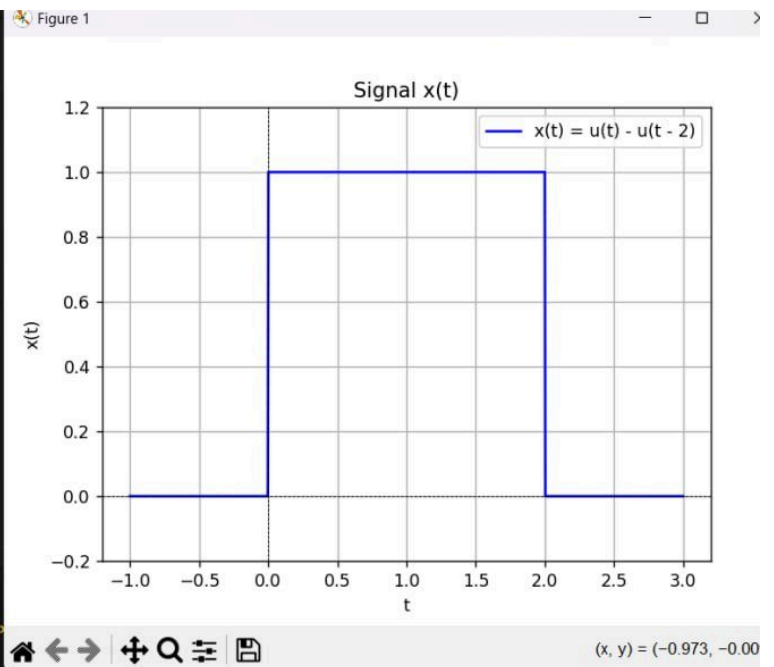
# محور زمان t تعریف دامنه
t = np.linspace(-1, 3, 1000) # از -1 تا 3 با 1000 نقطه

# 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()

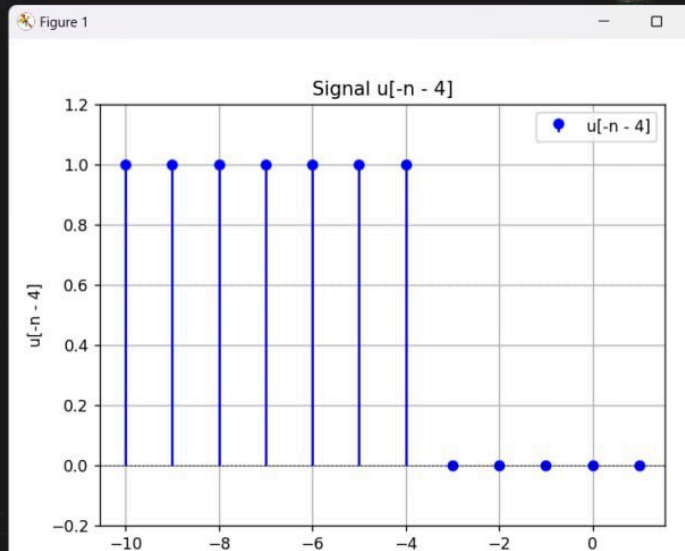
```



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور گسسته n تعریف دامنه
5 n = np.arange(-10, 2, 1) # از -10 تا 1
6
7 # تعریف تابع واحد گسسته u[n]
8 def u(n):
9     return np.where(n >= 0, 1, 0)
10
11 # تعریف سیگنال u[-n - 4]
12 x_n = u(-n - 4)
13
14 # رسم سیگنال
15 plt.figure()
16 plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='u[-n - 4]')
17 plt.xlabel('n')
18 plt.ylabel('u[-n - 4]')
19 plt.title('Signal u[-n - 4]')
20 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
21 plt.grid(True)
22 plt.ylim(-0.2, 1.2) # تنظیم محدوده محور عمودی
23 plt.legend()
24 plt.show()
25

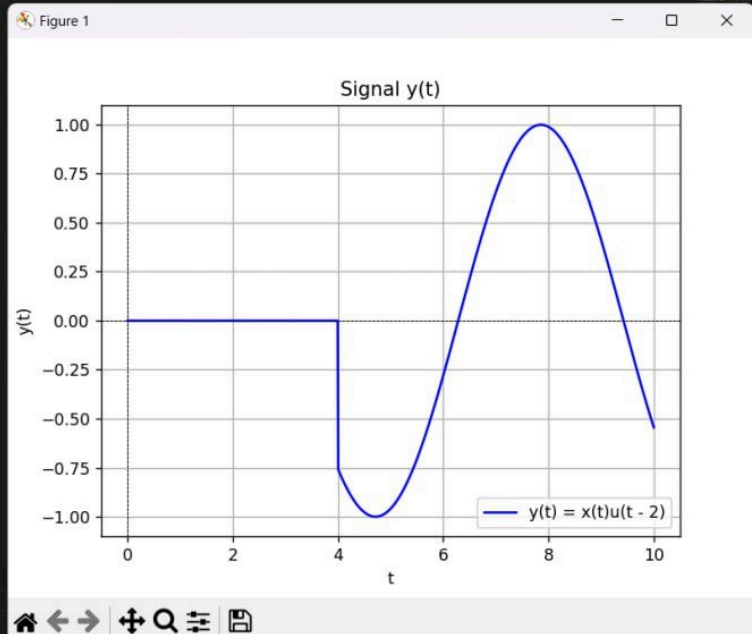
```



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور زمان t تعریف دامنه
5 t = np.linspace(0, 10, 1000) # از 0 تا 10 با 1000 نقطه
6
7 # تعریف تابع x(t)
8 x_t = np.sin(t) # را به عنوان سینوس تعریف کردیم x(t) تابع
9
10 # تعریف تابع واحد u(t-2)
11 def u(t):
12     return np.where(t >= 2, 1, 0)
13
14 # تعریف سیگنال y(t) = x(t) * u(t-2)
15 y_t = x_t * u(t - 2)
16
17 # رسم سیگنال
18 plt.figure()
19 plt.plot(t, y_t, label='y(t) = x(t)u(t - 2)', color='blue')
20 plt.xlabel('t')
21 plt.ylabel('y(t)')
22 plt.title('Signal y(t)')
23 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
24 plt.axvline(2, color='black', lw=0.5, ls='--') # خط عمودی صفر
25 plt.grid(True)
26 plt.legend()
27 plt.show()
28

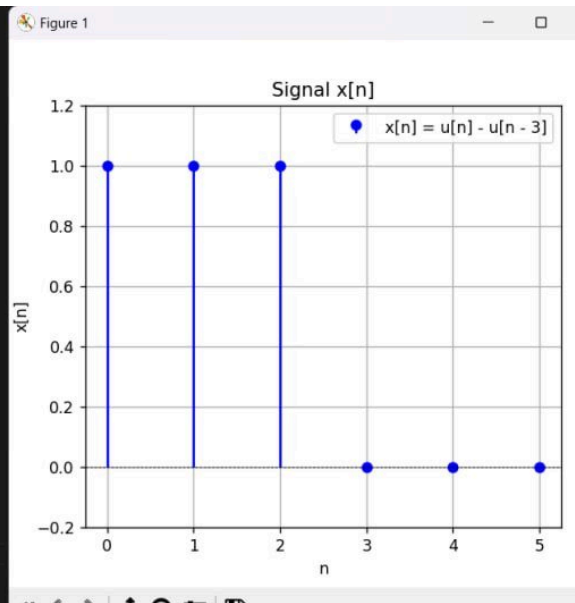
```




```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور گسسته n تعریف دامنه
5 n = np.arange(0, 6, 1) # 5 تا 0 از
6
7 # تعریف تابع واحد گسسته u[n]
8 def u(n):
9     return np.where(n >= 0, 1, 0)
10
11 # تعریف سیگنال x[n] = u[n] - u[n - 3]
12 x_n = u(n) - u(n - 3)
13
14 # رسم سیگنال
15 plt.figure()
16 plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='x[n] = u[n] - u[n - 3]')
17 plt.xlabel('n')
18 plt.ylabel('x[n]')
19 plt.title('Signal x[n]')
20 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
21 plt.grid(True)
22 plt.ylim(-0.2, 1.2) # تنظیم محدوده محور عمودی
23 plt.legend()
24 plt.show()
25

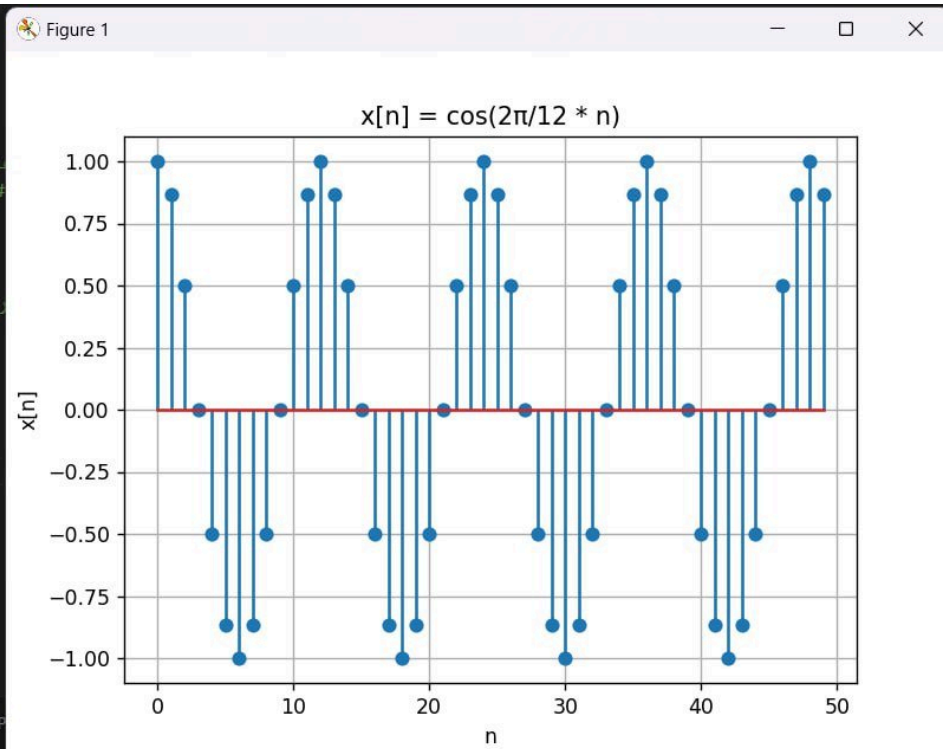
```



```

5.py > ...
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # محور گسسته n تعریف دامنه
5  n = np.arange(0, 50, 1) # تعداد صحیح
6  x_n = np.cos((2 * np.pi / 12) * n) #
7
8  # رسم سیگنال گسسته
9  plt.figure()
10 plt.stem(n, x_n) # رسم سیگنال گسسته
11 plt.xlabel('n')
12 plt.ylabel('x[n]')
13 plt.title('x[n] = cos(2π/12 * n)')
14 plt.grid(True)
15 plt.show()
16

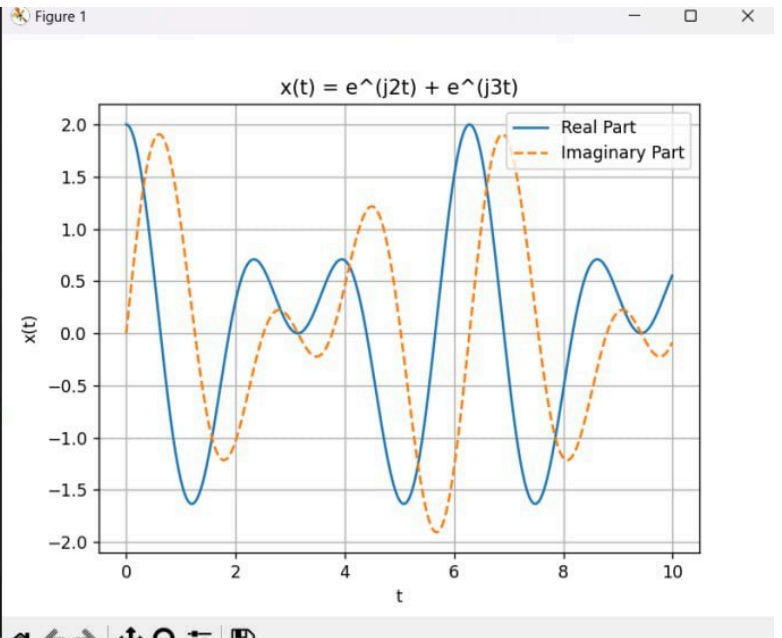
```



```

5.py > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور زمان t تعریف دامنه
5 t = np.arange(0, 10, 0.01) # 0 تا 10 با گام 0.01
6
7 # تعریف سیگنال مختلط x(t) = e^(j2t) + e^(j3t)
8 x_t = np.exp(1j * 2 * t) + np.exp(1j * 3 * t)
9
10 # استخراج قسمت حقیقی و موهومی
11 real_part = np.real(x_t)
12 imaginary_part = np.imag(x_t)
13
14 # رسم قسمت حقیقی سیگنال
15 plt.figure()
16 plt.plot(t, real_part, label='Real Part')
17 plt.plot(t, imaginary_part, label='Imaginary Part', linestyle='--')
18 plt.xlabel('t')
19 plt.ylabel('x(t)')
20 plt.title('x(t) = e^(j2t) + e^(j3t)')
21 plt.legend()
22 plt.grid(True)
23 plt.show()

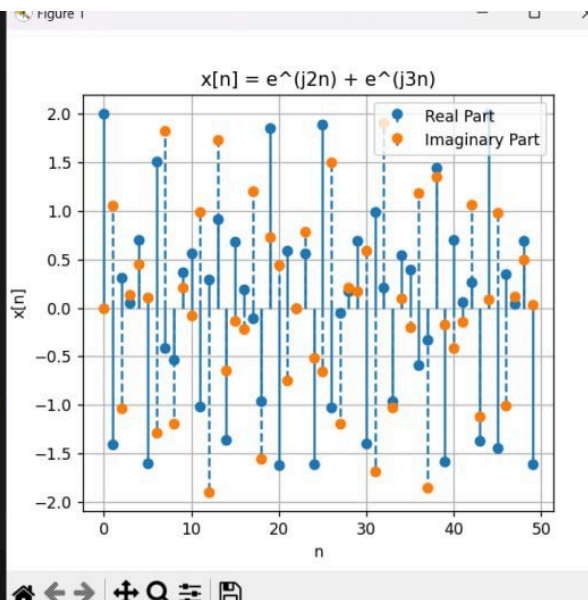
```



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور گسسته n تعریف دامنه
5 n = np.arange(0, 50, 1) # (اعداد صحیح) 1 با گام 50 تا 0 از
6
7 # تعریف سیگنال مختلط x[n] = e^(j2n) + e^(j3n)
8 x_n = np.exp(1j * 2 * n) + np.exp(1j * 3 * n)
9
10 # استخراج قسمت حقیقی و موهومی
11 real_part = np.real(x_n)
12 imaginary_part = np.imag(x_n)
13
14 # رسم قسمت حقیقی سیگنال
15 plt.figure()
16 plt.stem(n, real_part, label='Real Part', basefmt=" ")
17 plt.stem(n, imaginary_part, label='Imaginary Part', basefmt=" ", linefmt='--', markerfmt='o')
18 plt.xlabel('n')
19 plt.ylabel('x[n]')
20 plt.title('x[n] = e^(j2n) + e^(j3n)')
21 plt.legend()
22 plt.grid(True)
23 plt.show()

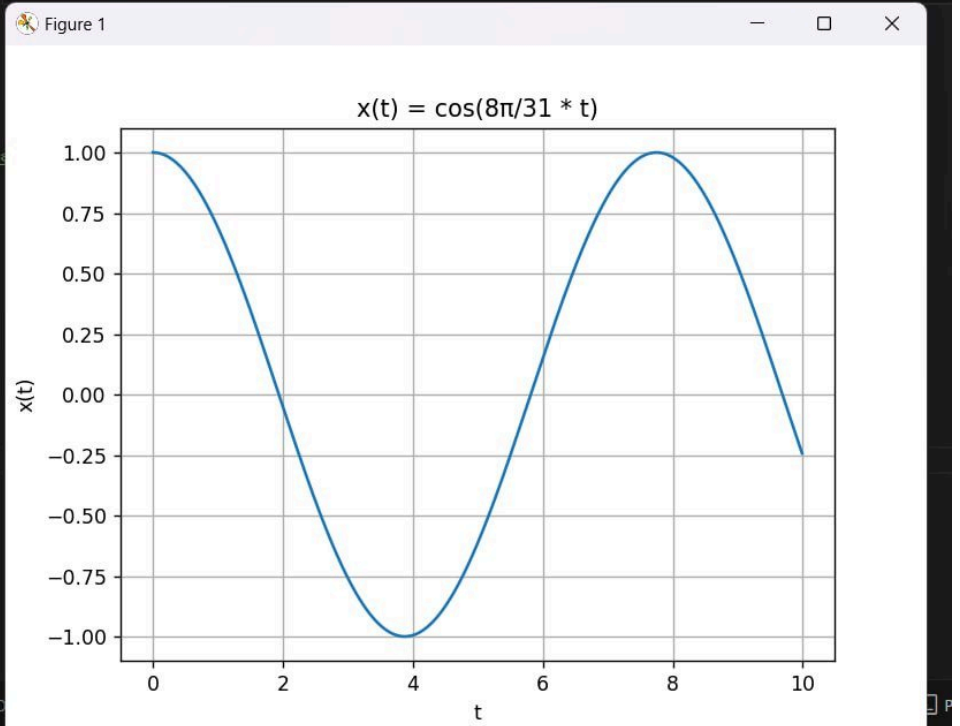
```



```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور زمان t تعریف دامنه
5 t = np.arange(0, 10, 0.01) # 0.01 نام
6 x_t = np.cos((8 * np.pi / 31) * t) #
7
8 # رسم سیگنال
9 plt.figure()
10 plt.plot(t, x_t)
11 plt.xlabel('t')
12 plt.ylabel('x(t)')
13 plt.title('x(t) = cos(8π/31 * t)')
14 plt.grid(True)
15 plt.show()
16

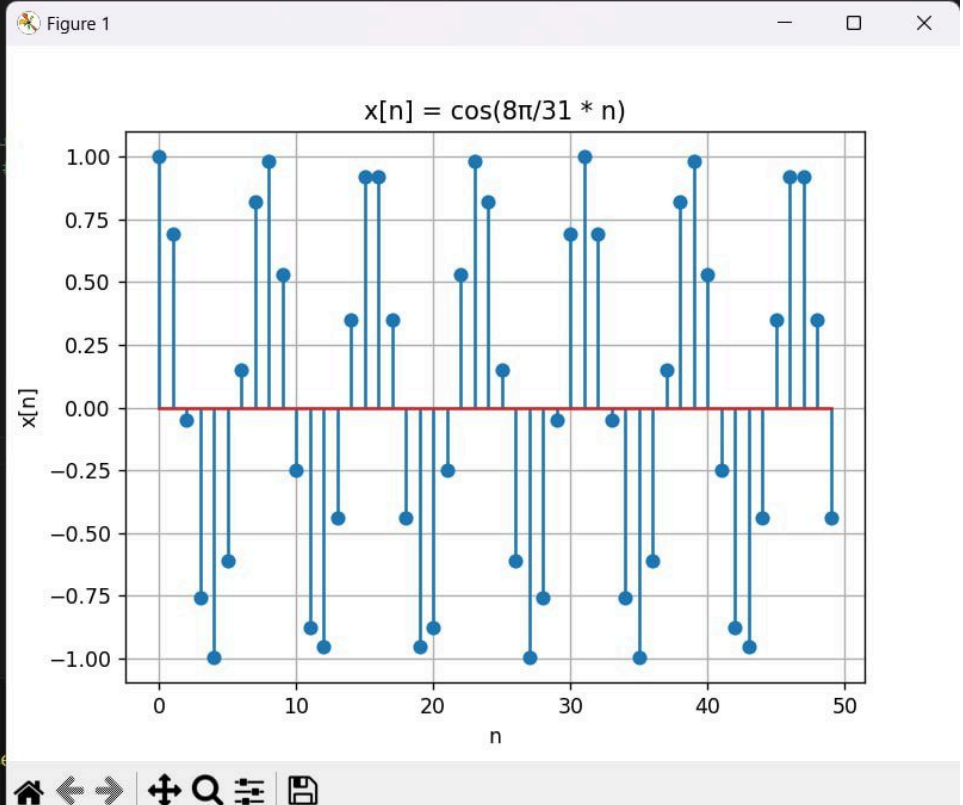
```



```

> رسم سیگنال 3.py > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور گسسته n تعریف دامنه
5 n = np.arange(0, 50, 1) # (مقادیر صحیح)
6 x_n = np.cos((8 * np.pi / 31) * n)
7
8 # رسم سیگنال گسسته
9 plt.figure()
10 plt.stem(n, x_n)
11 plt.xlabel('n')
12 plt.ylabel('x[n]')
13 plt.title('x[n] = cos(8π/31 * n)')
14 plt.grid(True)
15 plt.show()
16

```



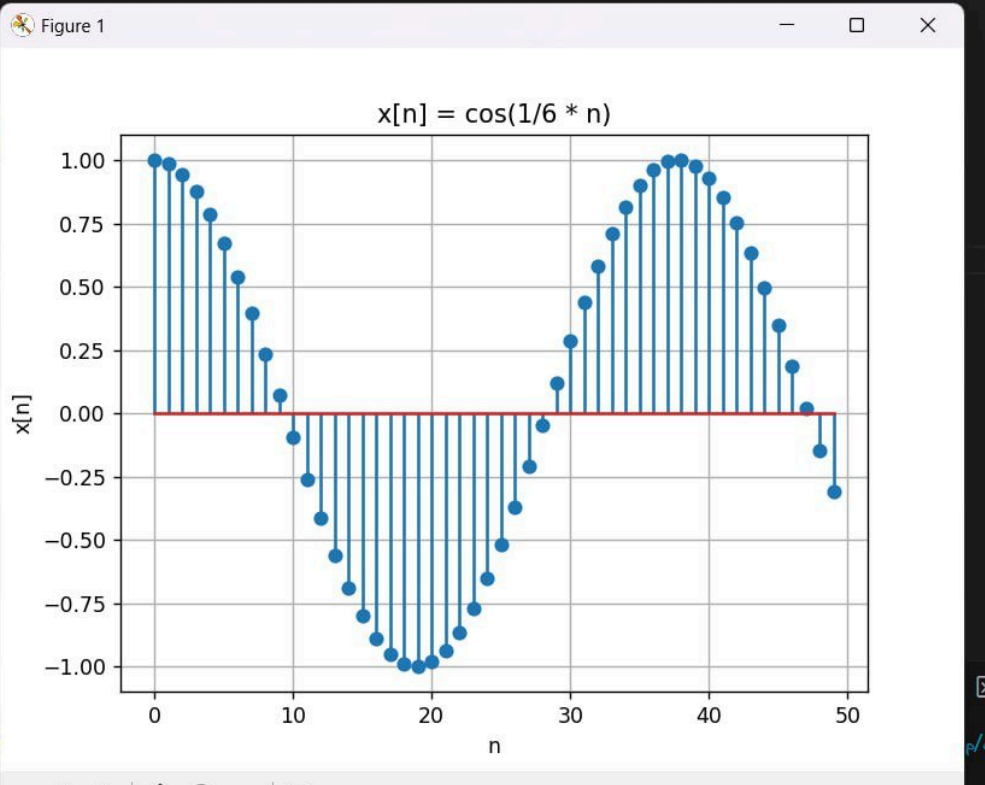
```

import numpy as np
import matplotlib.pyplot as plt

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

# 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()

```



```
Welcome 1.py x
1.py > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محاور زمان t تعریف دامنه
5 t = np.arange(0, 10, 0.01) # 0.01
6 x_t = np.cos((1/6) * t) # گنال
7
8 # رسم سیگنال
9 plt.figure()
10 plt.plot(t, x_t)
11 plt.xlabel('t')
12 plt.ylabel('x(t)')
13 plt.title('x(t) = cos(1/6 * t)')
14 plt.grid(True)
15 plt.show()
16
```

