<table>
<tr><td></td><td colspan="2">Marwadi University
Faculty of Engineering & Technology
Department of Information and Communication Technology</td></tr>
<tr><td>Subject: Programming With Python (01CT1309)</td><td colspan="2">Aim: Analysis of Discrete-Time Signals Using Z-Transform</td></tr>
<tr><td>Experiment No: 17</td><td>Date:</td><td>Enrollment No:92400133037</td></tr>
</table>

Aim: Analysis of Discrete-Time Signals Using Z-Transform

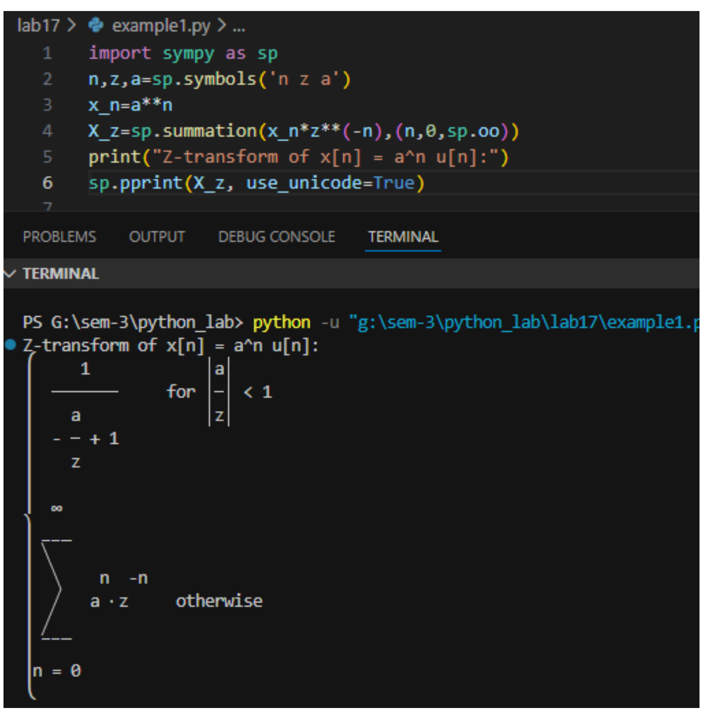
IDE:

Install Library

pip install sympy

Example 1:

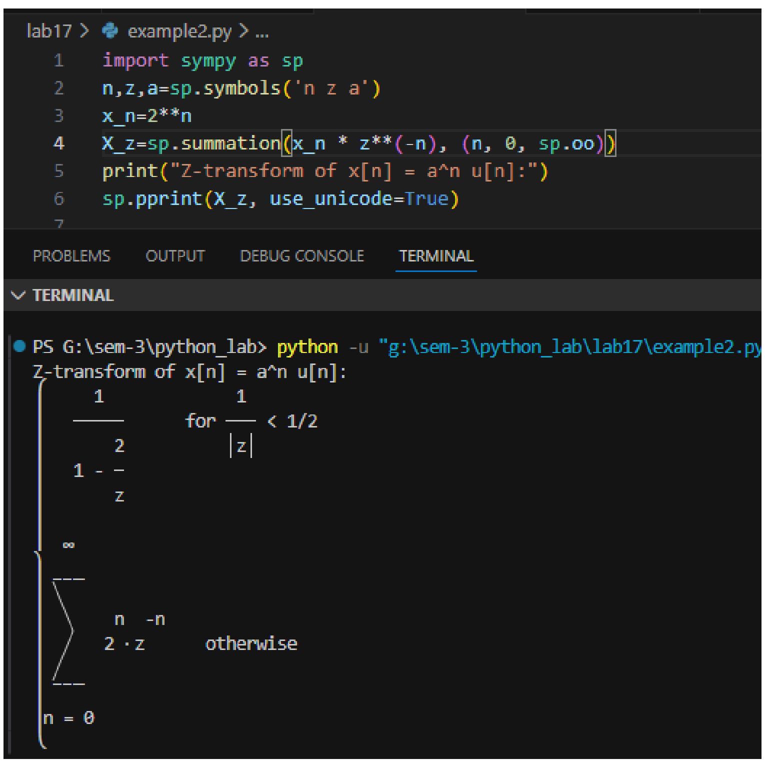
```
import sympy as sp
# Define symbols
n, z, a = sp.symbols('n z a')
# Define the signal x[n] = a^n * u[n]
x_n = a**n
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = a^n u[n]:")
sp.pprint(X_z, use_unicode=True)
```



Marwadi University (Marwadi Chandarana Group) NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Analysis of Discrete-Time Signals Using Z-Transform	
Experiment No: 17	**Date:**	**Enrollment No:92400133037**

Example 2:

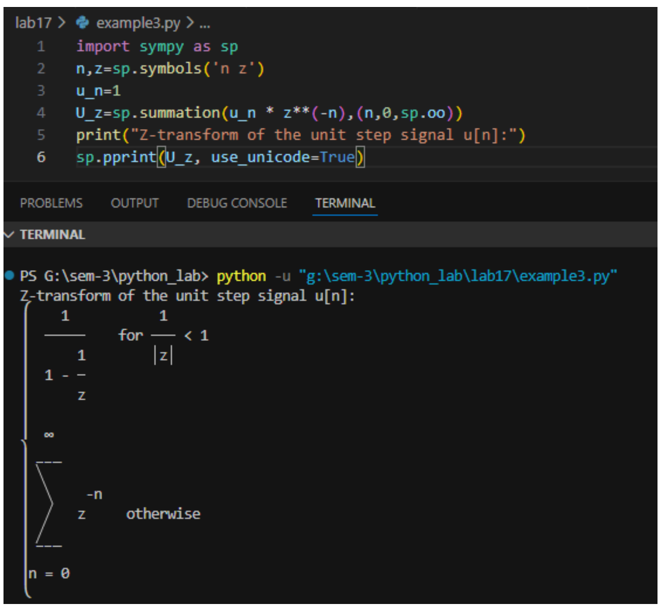
```
# Define symbols
n, z, a = sp.symbols('n z a')
# Define the signal x[n] = a^n * u[n]
x_n = 2**n
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = a^n u[n]:")
sp.pprint(X_z, use_unicode=True)
```



Example 3:

```
import sympy as sp
# Define symbols
n, z = sp.symbols('n z')
# Define the unit step signal u[n]
u_n = 1
```

```
# Compute the Z-transform
U_z = sp.summation(u_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of the unit step signal u[n]:")
sp.pprint(U_z, use_unicode=True)
```

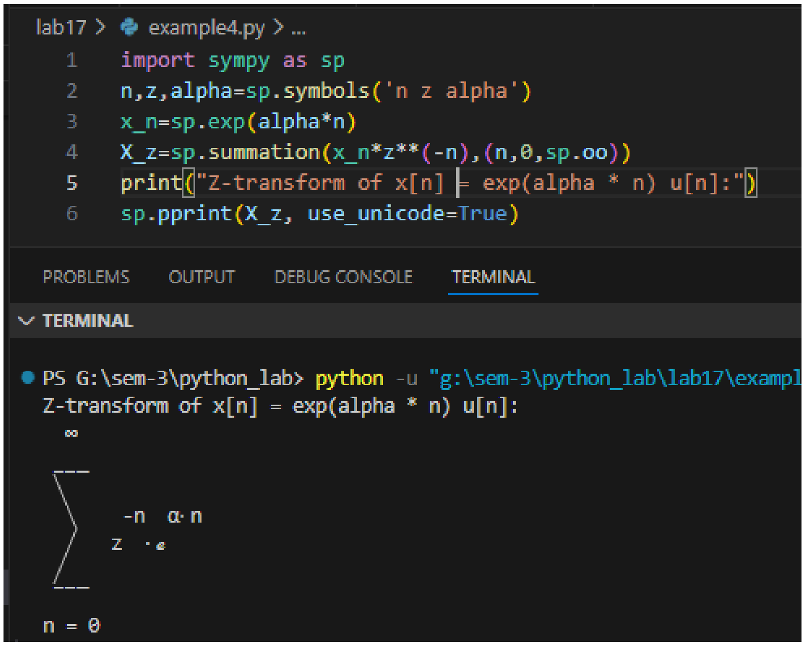


Example 4:

```
import sympy as sp
# Define symbols
n, z, alpha = sp.symbols('n z alpha')
# Define the signal x[n] = exp(alpha * n) * u[n]
x_n = sp.exp(alpha * n)
# Compute the Z-transform
X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))
# Print the result
print("Z-transform of x[n] = exp(alpha * n) u[n]:")
sp.pprint(X_z, use_unicode=True)
```

Marwadi University Marwadi Chandarana Group NAAC A+	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**
Subject: Programming With Python (01CT1309)	**Aim:** Analysis of Discrete-Time Signals Using Z-Transform

Experiment No: 17	**Date:**	**Enrollment No:92400133037**



Example 5:

```
import sympy as sp
# Define symbols
n, z = sp.symbols('n z')
# Define the finite sequence x[n] = {1, 2, 3}
x_n = [1, 2, 3]
# Compute the Z-transform manually
X_z = sum(x_n[i] * z**(-i) for i in range(len(x_n)))
# Print the result
print("Z-transform of the finite sequence {1, 2, 3}:")
sp.pprint(X_z, use_unicode=True)
```

Marwadi University	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Analysis of Discrete-Time Signals Using Z-Transform	
Experiment No: 17	**Date:**	**Enrollment No:92400133037**

```
lab17 > example5.py > ...
import sympy as sp
n,z=sp.symbols('n z')
x_n=[1, 2, 3]
X_z=sum(x_n[i]*z**(-i)for i in range(len(x_n)))
print("Z-transform of the finite sequence {1, 2, 3}:")
sp.pprint(X_z, use_unicode=True)
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

TERMINAL

PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example5.py"
Z-transform of the finite sequence {1, 2, 3}:
    2   3
1 + ─ + ──
    z    2
        z
```

Example 6

import sympy as sp

Define symbols

n, z, omega = sp.symbols('n z omega')

Define the sinusoidal sequence x[n] = sin(omega * n) * u[n]

x_n = sp.sin(omega * n)

Compute the Z-transform

X_z = sp.summation(x_n * z**(-n), (n, 0, sp.oo))

Print the result

print("Z-transform of x[n] = sin(omega * n) u[n]:")

sp.pprint(X_z, use_unicode=True)

```
lab17 > example6.py > ...
import sympy as sp
n,z,omega=sp.symbols('n z omega')
x_n=sp.sin(omega*n)
X_z=sp.summation(x_n *z**(-n),(n,0,sp.oo))
print("Z-transform of x[n] = sin(omega * n) u[n]:")
sp.pprint(X_z, use_unicode=True)
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

TERMINAL

PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\example6.py"
Z-transform of x[n] = sin(omega * n) u[n]:
 ∞
___
╲
 ╲    -n
 ╱   z  ⋅sin(n⋅ω)
╱
‾‾‾
n = 0
```

	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Analysis of Discrete-Time Signals Using Z-Transform	
Experiment No: 17	**Date:**	**Enrollment No:92400133037**

Post Lab Exercise:

- Using Python, compute the Z-transform of the sequence $x[n] = 3^n u[n]$.
- Using Python, compute the Z-transform of the sequence $x[n] = cos(wn)u[n]$.

```
lab17 > postLab.py > ...
import sympy as sp
n,z=sp.symbols('n z')
x_n=3**n
X_z=sp.summation(x_n*z**(-n),(n,0,sp.oo))
print("Z-transform of x[n]=3^n u[n]:")
sp.pprint(X_z,use_unicode=True)
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

TERMINAL
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\postLab.py"
Z-transform of x[n]=3^n u[n]:
⎧   1          1
⎪ ─────   for ─── < 1/3
⎪     3       |z|
⎪ 1 - ─
⎪     z
⎪
⎨ ∞
⎪ ___
⎪ ╲
⎪  ╲    n  -n
⎪  ╱   3 ⋅z       otherwise
⎪ ╱
⎪ ‾‾‾
⎩n = 0
```

```
#b
n,z,omega=sp.symbols('n z omega')
x_n=sp.cos(omega*n)
X_z=sp.summation(x_n*z**(-n),(n,0,sp.oo))
print("Z-transform of x[n] = cos(omega*n) u[n]:")
sp.pprint(X_z,use_unicode=True)
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

TERMINAL
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab17\postLab.py"
Z-transform of x[n] = cos(omega*n) u[n]:
 ∞
___
╲
 ╲    -n
 ╱   z  ⋅cos(n⋅ω)
╱
‾‾‾
n = 0
```

GITHUB LINK:
https://github.com/Heer972005/Python_Lab