```python
import numpy as np
import matplotlib.pyplot as plt
# Discrete time indices
n = np.arange(0, 21)
# Define input signals
delta_n = np.zeros(21)
delta_n[0] = 1
u_n = np.ones(21)
#  y[n] = 0.5*y[n-1] + x[n]
a1 = 0.5
b0 = 1.0
# Initialize output arrays
h_n = np.zeros(21)
s_n = np.zeros(21)
# Simulate impulse response h[n]
for i in range(21):
    if i == 0:
        h_n[i] = b0 * delta_n[i]
    else:
        h_n[i] = a1 * h_n[i-1] + b0 * delta_n[i]
# Simulate step response s[n]
for i in range(21):
    if i == 0:
        s_n[i] = b0 * u_n[i]
    else:
        s_n[i] = a1 * s_n[i-1] + b0 * u_n[i]
# Plotting the results

#IMPULSE RESPONSE PLOT
plt.figure(figsize=(12, 8))
plt.subplot(2, 1, 1)
plt.stem(n, h_n, linefmt='b-', markerfmt='bo', basefmt=' ', label='Impulse Response h[n]')
plt.title('Impulse Response of the System')
plt.xlabel('n (Discrete Time Index)')
plt.ylabel('h[n]')
plt.legend()
plt.grid()

#STEP RESPONSE PLOT
plt.subplot(2, 1, 2)
plt.stem(n, s_n, linefmt='r-', markerfmt='ro', basefmt=' ', label='Step Response s[n]')
plt.title('Step Response of the System')
plt.xlabel('n (Discrete Time Index)')
plt.ylabel('s[n]')
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

Impulse Response of the System

h[n]

n (Discrete Time Index)

Impulse Response h[n]

Step Response of the System

s[n]

n (Discrete Time Index)

Step Response s[n]