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


# Intergrated Function
def f1(x):
    return x ** 3


def f2(x):
    return np.sinc(x / np.pi)


def f3(x):
    return np.sin(x ** 2)


# Romberg Intergration
def romberg(f, a, b, tol=1e-12, max_level=5):
    T = np.zeros((max_level, max_level), dtype=float)
    T[0, 0] = 0.5 * (b - a) * (f(a) + f(b))
    for k in range(1, max_level):
        n_new = 2 ** k
        h = (b - a) / n_new
        m_count = 2 ** (k - 1)
        xs = a + (2 * np.arange(1, m_count + 1) - 1) * h
        T[k, 0] = 0.5 * T[k - 1, 0] + h * np.sum(f(xs))
        for j in range(1, k + 1):
            T[k, j] = T[k, j - 1] + (T[k, j - 1] - T[k - 1, j - 1]) / (4 ** j - 1)
        if k > 0 and abs(T[k, k] - T[k - 1, k - 1]) < tol:
            return T[:k + 1, :k + 1]
    return T


# Parameters
functions = [f1, f2, f3]
a_values = [6.0, 0.0, 0.0]
b_values = [100.0, 1.0, 1.0]
names = ["x^3", "sin(x)/x", "sin(x^2)"]

for idx, f in enumerate(functions):
    print(f"\n=== Romberg Integration: ∫{names[idx]} dx [{a_values[idx]}, {b_values[idx]}] ===")
    T = romberg(f, a_values[idx], b_values[idx], tol=1e-12, max_level=5)
    n = T.shape[0]

    # Output
    labels = ["T", "S", "C", "K", "L"]
```

```
48      print("Romberg Table:")
49      for j in range(n):
50          row_label = labels[j] if j < len(labels) else f"R{
    j}"
51          vals = [f"{T[k, j]:16.8f}" for k in range(j, n)]
52          print(f"{row_label}: " + " ".join(vals))
53
54      best = T[n - 1, n - 1]
55      print(f"Final Value = {best:16.12f}")
56
57      # Plot
58      cell_text = []
59      row_labels = [labels[j] for j in range(n)]
60      col_labels = [str(2 ** k) for k in range(n)]
61      for j in range(n):
62          row = [f"{T[k, j]:16.8f}" for k in range(j, n
    )] + [""] * j
63          cell_text.append(row)
64
65      fig, ax = plt.subplots(figsize=(2.0 * n + 1.5, 1.0 * n
     + 1.2))
66      ax.axis('off')
67      ax.set_title(f"Romberg Table: {names[idx]}", fontsize=
    12)
68      the_table = ax.table(cellText=cell_text,
69                           rowLabels=row_labels,
70                           colLabels=col_labels,
71                           cellLoc='left',
72                           loc='center')
73      the_table.auto_set_font_size(False)
74      the_table.set_fontsize(10)
75      the_table.scale(1.3, 1.3)
76      for (r, c), cell in the_table.get_celld().items():
77          cell.set_edgecolor('black')
78          if cell.get_text().get_text() == "":
79              cell.set_facecolor('#f0f0f0')
80
81      plt.tight_layout()
82
83      # Auto Saving
84      filename = f"Romberg_{names[idx].replace('/', '_')}.
    png"
85      plt.savefig(filename, dpi=300)
86      print(f"Saved: {filename}")
87
88      plt.show()
89
90
```