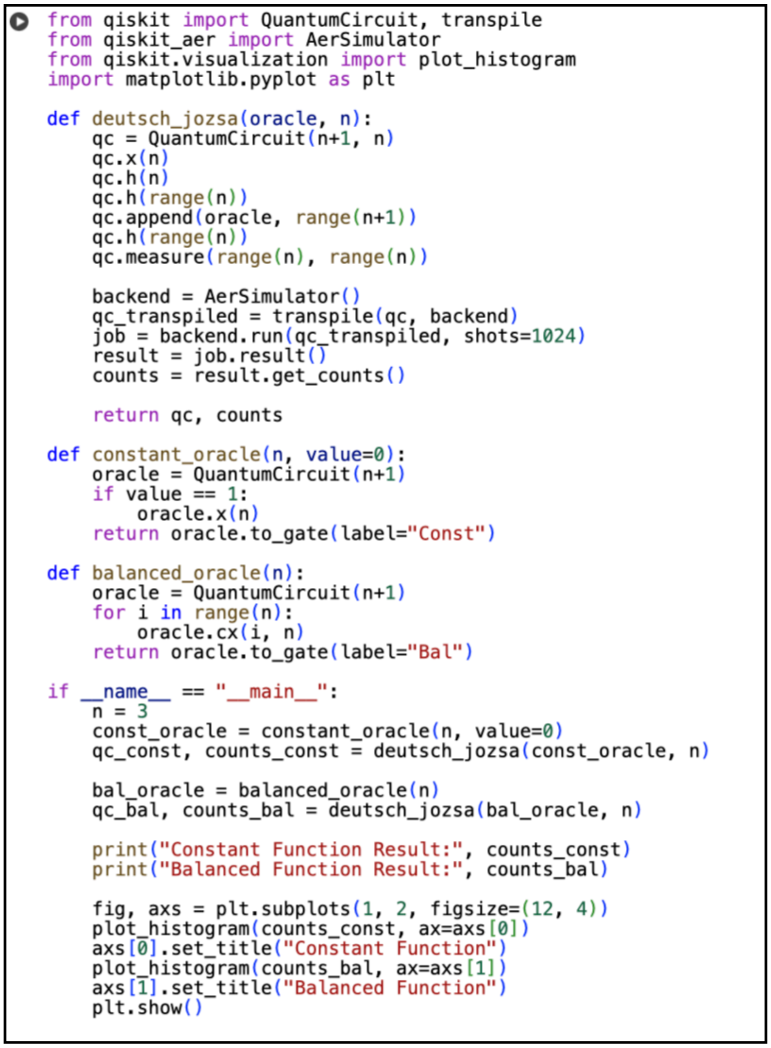
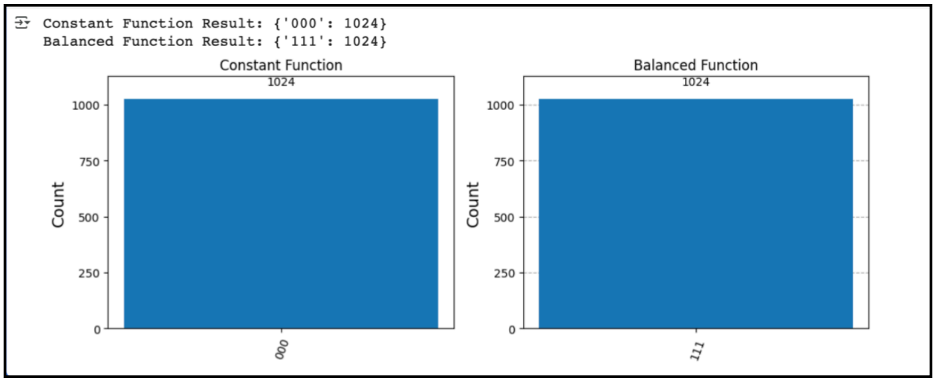
Experiment no. 6

Program:



```
from qiskit import QuantumCircuit, transpile
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
import matplotlib.pyplot as plt

def deutsch_jozsa(oracle, n):
    qc = QuantumCircuit(n+1, n)
    qc.x(n)
    qc.h(n)
    qc.h(range(n))
    qc.append(oracle, range(n+1))
    qc.h(range(n))
    qc.measure(range(n), range(n))

    backend = AerSimulator()
    qc_transpiled = transpile(qc, backend)
    job = backend.run(qc_transpiled, shots=1024)
    result = job.result()
    counts = result.get_counts()

    return qc, counts

def constant_oracle(n, value=0):
    oracle = QuantumCircuit(n+1)
    if value == 1:
        oracle.x(n)
    return oracle.to_gate(label="Const")

def balanced_oracle(n):
    oracle = QuantumCircuit(n+1)
    for i in range(n):
        oracle.cx(i, n)
    return oracle.to_gate(label="Bal")

if __name__ == "__main__":
    n = 3
    const_oracle = constant_oracle(n, value=0)
    qc_const, counts_const = deutsch_jozsa(const_oracle, n)

    bal_oracle = balanced_oracle(n)
    qc_bal, counts_bal = deutsch_jozsa(bal_oracle, n)

    print("Constant Function Result:", counts_const)
    print("Balanced Function Result:", counts_bal)

    fig, axs = plt.subplots(1, 2, figsize=(12, 4))
    plot_histogram(counts_const, ax=axs[0])
    axs[0].set_title("Constant Function")
    plot_histogram(counts_bal, ax=axs[1])
    axs[1].set_title("Balanced Function")
    plt.show()
```

Output:



```
Constant Function Result: {'000': 1024}
Balanced Function Result: {'111': 1024}
```