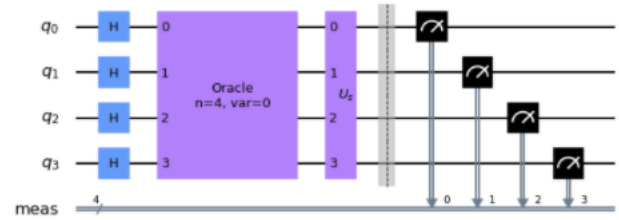
Name: Okoye, Adunife Kizito
Student ID: 100611918
CSCI 4140U

Assignment Two
EXPERIMENT ONE

In [34]:
```
#Problem One
n = 4
variant = 0
oracle = grover_problem_oracle(n, variant, print_solutions = True)
qc = QuantumCircuit(n)
qc = initialize_s(qc, [0,1,2,3])
qc.append(oracle, [0,1,2,3])
qc.append(diffuser(n), [0,1,2,3])
qc.measure_all()
qc.draw('mpl')
```

```
Solutions:
|1101>
```

Out[34]:

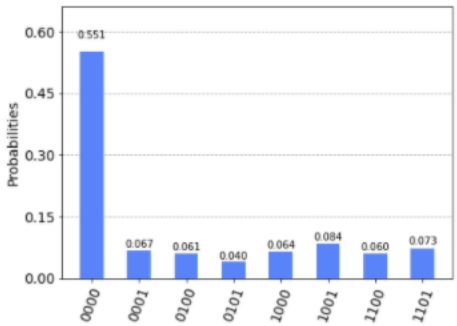


Problem 1 Histogram without noise;

In [54]:
```
results = execute(qc, backend=backend, shots=1024).result().get_counts()
print(results)
plot_histogram(results)
```

```
{'0000': 564, '0001': 69, '0100': 62, '0101': 41, '1000': 66, '1001': 86, '1100': 61, '1101': 75}
```

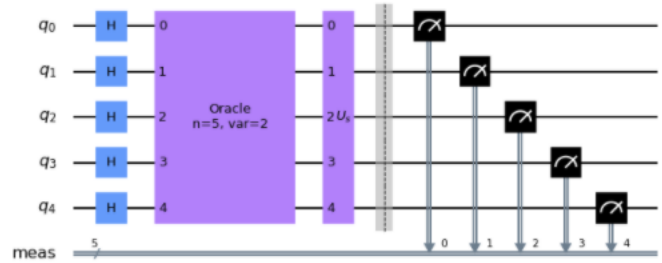
Out[54]:



In [25]:
```
#Problem Two
n = 5
variant = 2
oracle = grover_problem_oracle(n, variant, print_solutions = True)
qc = QuantumCircuit(n)
qc = initialize_s(qc, [0,1,2,3,4])
qc.append(oracle, [0,1,2,3,4])
qc.append(diffuser(n), [0,1,2,3,4])
qc.measure_all()
qc.draw('mpl')
```

```
Solutions:
|00100>
```

Out[25]:

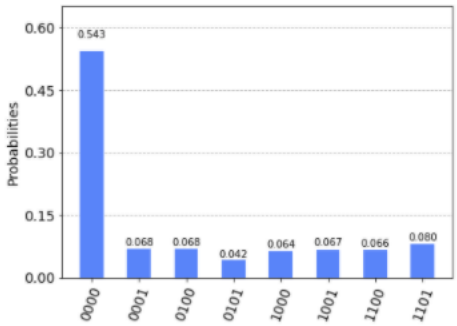


Problem 2 Histogram without noise;

```
In [56]: results = execute(qc, backend=backend, shots=1024).result().get_counts()
         print(results)
         plot_histogram(results)
```

```
{'0000': 556, '0001': 70, '0100': 70, '0101': 43, '1000': 66, '1001': 69, '1100': 68, '1101': 82}
```

Out[56]:



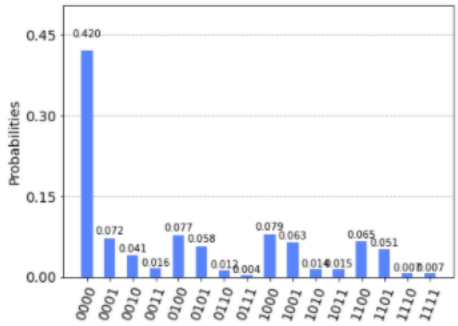
Problem	Gate Error				
No. 1	0.01	0.02	0.05	0.037	0.011
No. 2	0.01	0.02	0.05	0.07	0.021

Problem 1 Histograms with noise;

```
In [66]: noise_model = get_noise(0.01, 0.01)
         counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
         print(counts)
         plot_histogram(counts)
```

```
{'0000': 430, '0001': 74, '0010': 42, '0011': 16, '0100': 79, '0101': 59, '0110': 12, '0111': 4, '1000': 81, '1001':
65, '1010': 14, '1011': 15, '1100': 67, '1101': 52, '1110': 7, '1111': 7}
```

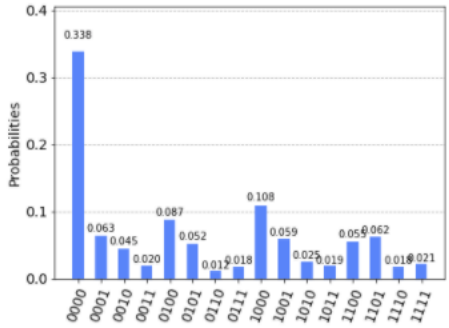
Out[66]:




```
In [67]: noise_model = get_noise(0.01, 0.02)
         counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
         print(counts)
         plot_histogram(counts)
```

```
{'0000': 346, '0001': 65, '0010': 46, '0011': 20, '0100': 89, '0101': 53, '0110': 12, '0111': 18, '1000': 111, '100
1': 60, '1010': 26, '1011': 19, '1100': 56, '1101': 63, '1110': 18, '1111': 22}
```

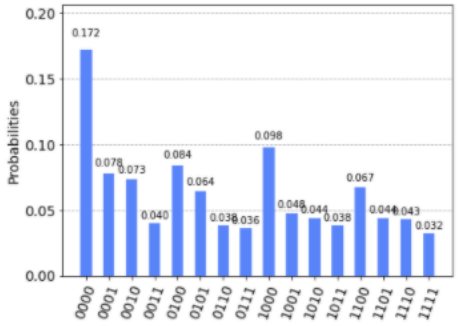
Out[67]:



In [68]:
```
noise_model = get_noise(0.01, 0.05)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 176, '0001': 80, '0010': 75, '0011': 41, '0100': 86, '0101': 66, '0110': 39, '0111': 37, '1000': 100, '100
1': 49, '1010': 45, '1011': 39, '1100': 69, '1101': 45, '1110': 44, '1111': 33}
```

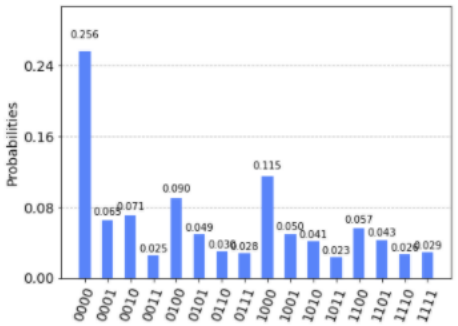
Out[68]:



In [69]:
```
noise_model = get_noise(0.01, 0.037)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 262, '0001': 67, '0010': 73, '0011': 26, '0100': 92, '0101': 50, '0110': 31, '0111': 29, '1000': 118, '100
1': 51, '1010': 42, '1011': 24, '1100': 58, '1101': 44, '1110': 27, '1111': 30}
```

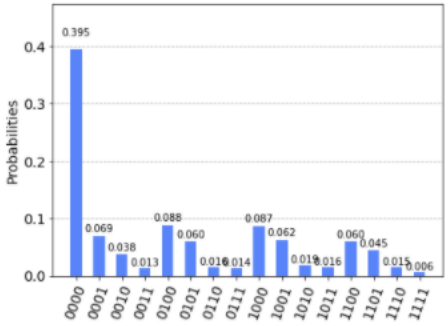
Out[69]:



In [74]:
```
noise_model = get_noise(0.01, 0.011)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 404, '0001': 71, '0010': 39, '0011': 13, '0100': 90, '0101': 61, '0110': 16, '0111': 14, '1000': 89, '1001':
64, '1010': 19, '1011': 16, '1100': 61, '1101': 46, '1110': 15, '1111': 6}
```

Out[74]:

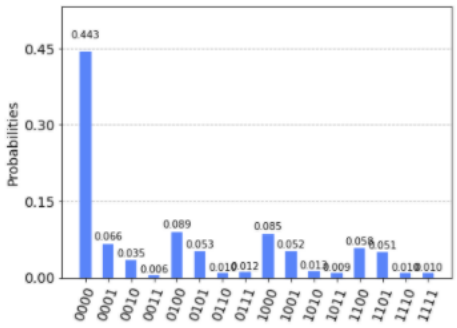


Problem 2 Histograms with noise;

In [75]:
```
noise_model = get_noise(0.01, 0.01)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 454, '0001': 68, '0010': 36, '0011': 6, '0100': 91, '0101': 54, '0110': 10, '0111': 12, '1000': 87, '1001': 53, '1010': 13, '1011': 9, '1100': 59, '1101': 52, '1110': 10, '1111': 10}
```

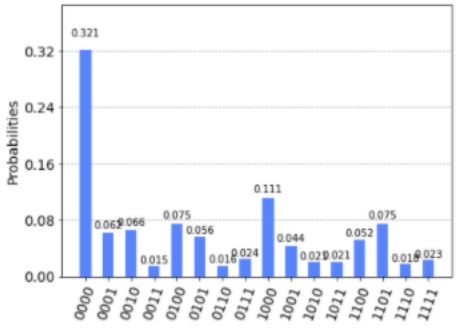
Out[75]:



In [76]:
```
noise_model = get_noise(0.01, 0.02)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 329, '0001': 63, '0010': 68, '0011': 15, '0100': 77, '0101': 57, '0110': 16, '0111': 25, '1000': 114, '1001': 45, '1010': 21, '1011': 22, '1100': 53, '1101': 77, '1110': 18, '1111': 24}
```

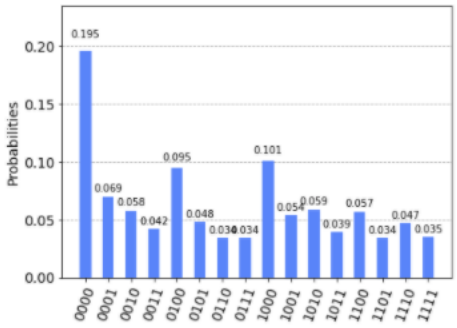
Out[76]:



In [77]:
```
noise_model = get_noise(0.01, 0.05)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 200, '0001': 71, '0010': 59, '0011': 43, '0100': 97, '0101': 49, '0110': 35, '0111': 35, '1000': 103, '1001': 55, '1010': 60, '1011': 40, '1100': 58, '1101': 35, '1110': 48, '1111': 36}
```

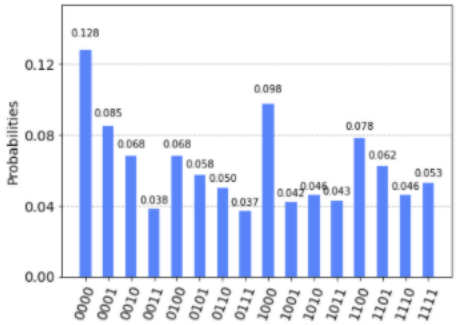
Out[77]:



In [78]:
```
noise_model = get_noise(0.01, 0.07)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 131, '0001': 87, '0010': 70, '0011': 39, '0100': 70, '0101': 59, '0110': 51, '0111': 38, '1000': 100, '100
1': 43, '1010': 47, '1011': 44, '1100': 80, '1101': 64, '1110': 47, '1111': 54}
```

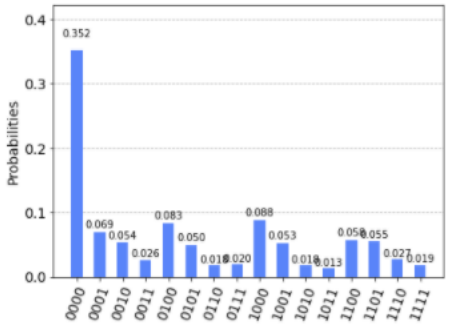
Out[78]:



In [79]:
```
noise_model = get_noise(0.01, 0.021)
counts = execute(qc, backend, noise_model=noise_model).result().get_counts()
print(counts)
plot_histogram(counts)
```

```
{'0000': 360, '0001': 71, '0010': 55, '0011': 27, '0100': 85, '0101': 51, '0110': 18, '0111': 20, '1000': 90, '1001':
54, '1010': 18, '1011': 13, '1100': 59, '1101': 56, '1110': 28, '1111': 19}
```

Out[79]:



Start with a noise model with 0.01 as the values for both the measurement error and the gate error. What impact does this have on the result of Grover's algorithm?

To my understanding and observations, the impact the noise model has on the Grover's algorithms is that the noise model generates more bits(*x-axis*) and lowers the value of the probabilities(*y*-axis).

EXPERIMENT TWO

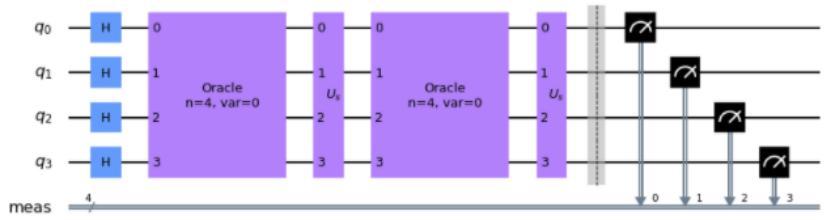
Hypothesis: noise decreases the probability of obtaining the correct result, so performing more iterations of Grover's algorithm will produce a better result.

Problem 1 – Two iterations

In [57]:
```
#Problem One
## Iteration X2
n = 4
variant = 0
oracle = grover_problem_oracle(n, variant, print_solutions = True)
qc = QuantumCircuit(n)
qc = initialize_s(qc, [0,1,2,3])
qc.append(oracle, [0,1,2,3])
qc.append(diffuser(n), [0,1,2,3])
qc.append(oracle, [0,1,2,3])
qc.append(diffuser(n), [0,1,2,3])
qc.measure_all()
qc.draw('mpl')
```

```
Solutions:
|1101>
```

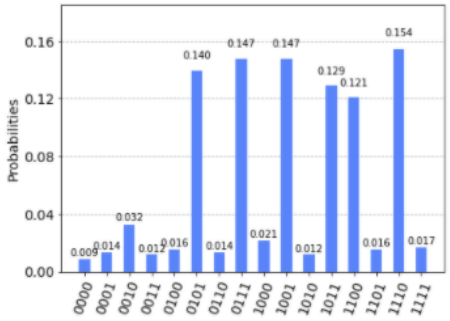
Out[57]:



In [58]:
```
results = execute(qc, backend=backend, shots=1024).result().get_counts()
print(results)
plot_histogram(results)
```

```
{'0000': 9, '0001': 14, '0010': 33, '0011': 12, '0100': 16, '0101': 143, '0110': 14, '0111': 151, '1000': 22, '1001':
151, '1010': 12, '1011': 132, '1100': 124, '1101': 16, '1110': 158, '1111': 17}
```

Out[58]:

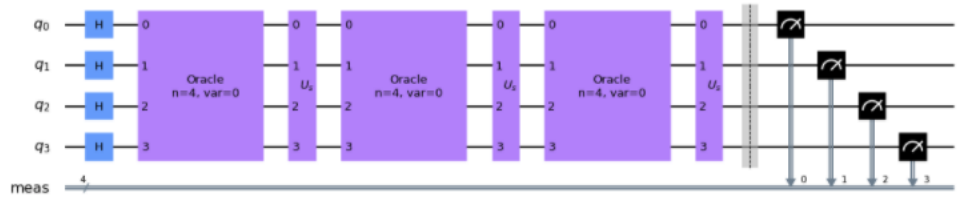


Problem 1 – Three iterations

In [59]:
```
## Iteration x3
n = 4
variant = 0
oracle = grover_problem_oracle(n, variant, print_solutions = True)
qc = QuantumCircuit(n)
qc = initialize_s(qc, [0,1,2,3])
qc.append(oracle, [0,1,2,3])
qc.append(diffuser(n), [0,1,2,3])
qc.append(oracle, [0,1,2,3])
qc.append(diffuser(n), [0,1,2,3])
qc.append(oracle, [0,1,2,3])
qc.append(diffuser(n), [0,1,2,3])
qc.measure_all()
qc.draw('mpl')
```

```
Solutions:
|1101>
```

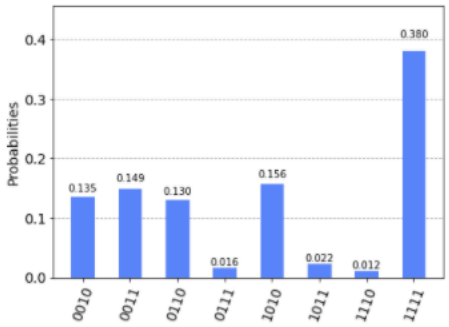
Out[59]:



In [60]:
```
results = execute(qc, backend=backend, shots=1024).result().get_counts()
print(results)
plot_histogram(results)
```

```
{'0010': 138, '0011': 153, '0110': 133, '0111': 16, '1010': 160, '1011': 23, '1110': 12, '1111': 389}
```

Out[60]:

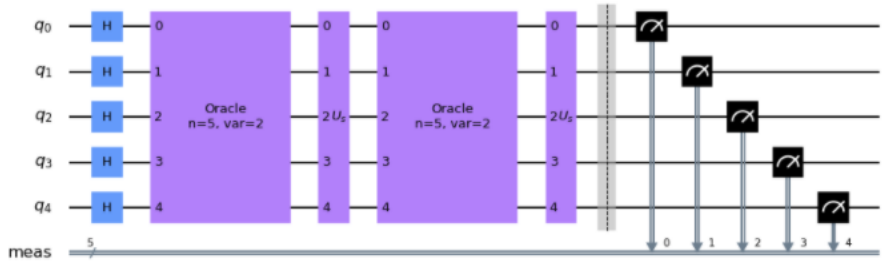


Problem 2 – Two iterations

In [61]:
```
#Problem Two
## Iteration X2
n = 5
variant = 2
oracle = grover_problem_oracle(n, variant, print_solutions = True)
qc = QuantumCircuit(n)
qc = initialize_s(qc, [0,1,2,3,4])
qc.append(oracle, [0,1,2,3,4])
qc.append(diffuser(n), [0,1,2,3,4])
qc.append(oracle, [0,1,2,3,4])
qc.append(diffuser(n), [0,1,2,3,4])
qc.measure_all()
qc.draw('mpl')
```

```
Solutions:
|00100>
```

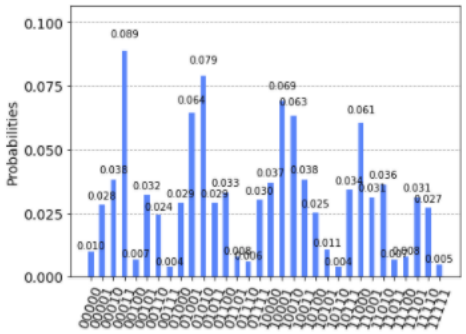
Out[61]:



In [62]:
```
results = execute(qc, backend=backend, shots=1024).result().get_counts()
print(results)
plot_histogram(results)
```

```
{'00000': 10, '00001': 29, '10000': 38, '10001': 71, '10010': 65, '10011': 39, '10100': 26, '10101': 11, '10110': 4,
'10111': 35, '11000': 62, '11001': 32, '11010': 37, '11011': 7, '11100': 8, '11101': 32, '11110': 28, '11111': 5, '00
010': 39, '00011': 91, '00100': 7, '00101': 33, '00110': 25, '00111': 4, '01000': 30, '01001': 66, '01010': 81, '0101
1': 30, '01100': 34, '01101': 8, '01110': 6, '01111': 31}
```

Out[62]:

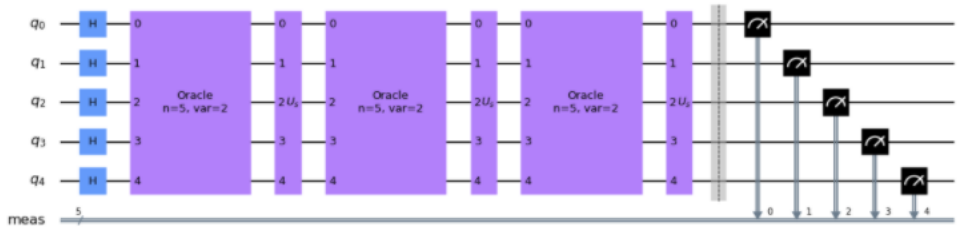


Problem 2 – Three iterations

In [63]:
```
## Iteration X3
n = 5
variant = 2
oracle = grover_problem_oracle(n, variant, print_solutions = True)
qc = QuantumCircuit(n)
qc = initialize_s(qc, [0,1,2,3,4])
qc.append(oracle, [0,1,2,3,4])
qc.append(diffuser(n), [0,1,2,3,4])
qc.append(oracle, [0,1,2,3,4])
qc.append(diffuser(n), [0,1,2,3,4])
qc.append(oracle, [0,1,2,3,4])
qc.append(diffuser(n), [0,1,2,3,4])
qc.measure_all()
qc.draw('mpl')
```

```
Solutions:
|00100>
```

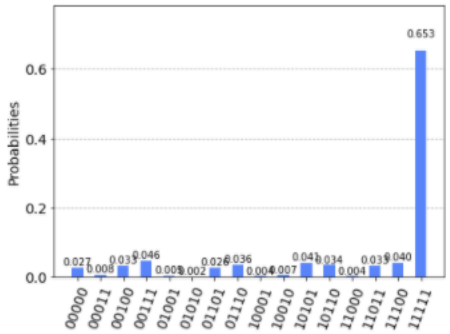
Out[63]:



In [64]:
```
results = execute(qc, backend=backend, shots=1024).result().get_counts()
print(results)
plot_histogram(results)
```

```
{'00000': 28, '10001': 4, '10010': 7, '10101': 42, '10110': 35, '11000': 4, '11011': 34, '11100': 41, '11111': 669,
'00011': 8, '00100': 34, '00111': 47, '01001': 5, '01010': 2, '01101': 27, '01110': 37}
```

Out[64]:



Is the hypothesis true?

I don't think the whole hypothesis is true. Although, certain error gate values in the noise model gives high probabilities the ones used for this experiment give low probabilities and do not give the correct exact result. Then performing iterations of Grover's algorithm did not really yield better results based on my observation the histograms.

Why do you think this is the case?

I think this is the case because the increase in number of iterations does not, affect the probabilities in the desired way.