Student Information

Name: Kaan Karaçanta

ID: 2448546

Part 1

a)

$2|\psi\rangle\langle\psi| - I$ where $|\psi\rangle = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$ and I is the identity matrix, and the inversion operation matrix is the following:

$$\frac{1}{2}\begin{bmatrix}-1 & 1 & 1 & 1\\ 1 & -1 & 1 & 1\\ 1 & 1 & -1 & 1\\ 1 & 1 & 1 & -1\end{bmatrix}$$

b)

To apply the matrix on the input, we need to multiply them.

$$\frac{1}{2}\begin{bmatrix}-1 & 1 & 1 & 1\\ 1 & -1 & 1 & 1\\ 1 & 1 & -1 & 1\\ 1 & 1 & 1 & -1\end{bmatrix}\begin{bmatrix}15\\23\\18\\32\end{bmatrix} = \frac{1}{2}\begin{bmatrix}58\\42\\52\\24\end{bmatrix} = \begin{bmatrix}29\\21\\26\\12\end{bmatrix}$$

Then, when we multiply the matrix again with the result, we get the following:

$$\frac{1}{2}\begin{bmatrix}-1 & 1 & 1 & 1\\ 1 & -1 & 1 & 1\\ 1 & 1 & -1 & 1\\ 1 & 1 & 1 & -1\end{bmatrix}\begin{bmatrix}29\\21\\26\\12\end{bmatrix} = \frac{1}{2}\begin{bmatrix}30\\46\\36\\64\end{bmatrix} = \begin{bmatrix}15\\23\\18\\32\end{bmatrix}$$

As it can be seen, we get the initial input again. This is because the matrix is the inversion operation matrix, inverts the inputs around their means, and when we apply it twice, it is the same as not applying it at all.

c)

My student id is 2448546 and its sum 33 is 100001 in binary. Hence, the oracle function is the following:

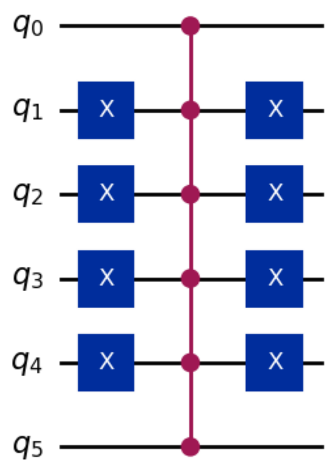

Figure 1: Oracle function

The Grover operator is the following:

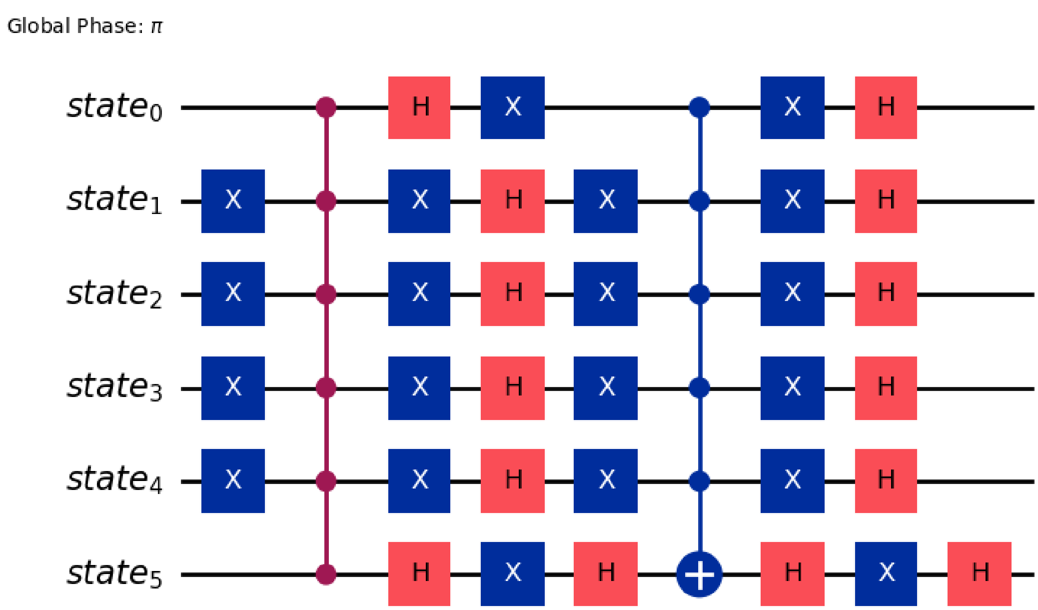

Figure 2: Grover operator

The final circuit is the following:

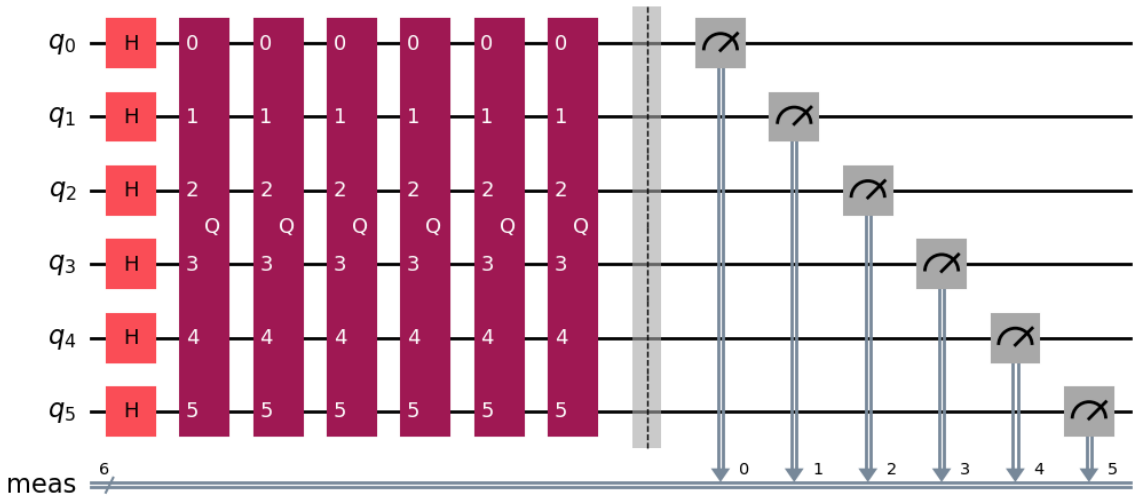

Figure 3: Final circuit

The result of the circuit is the following:

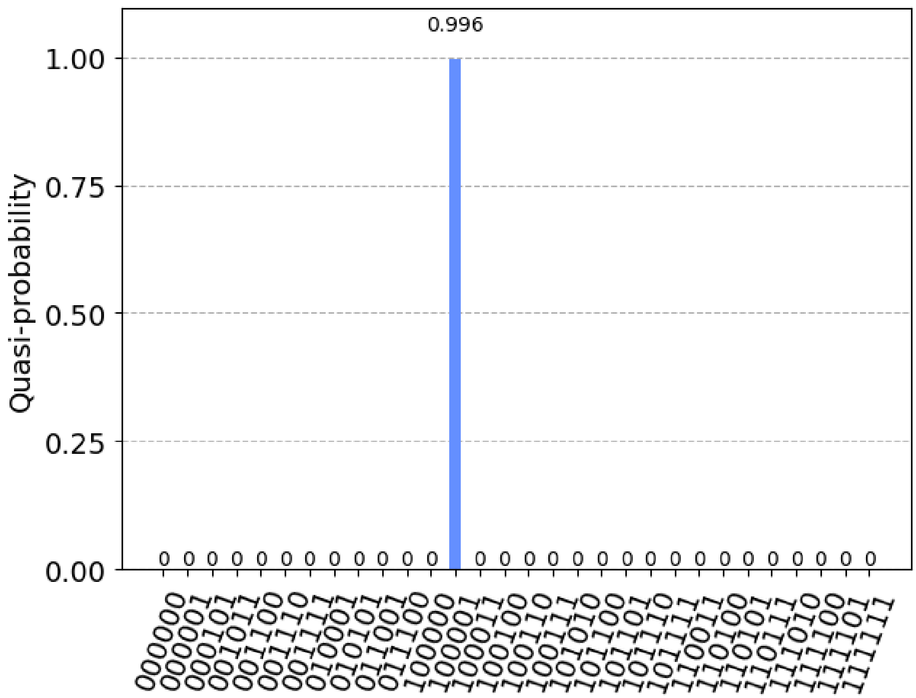

Figure 4: Final plot of the distributions

The plot shows a single dominant peak, with a quasi-probability very close to 1 (labeled as 0.996 on the plot), which is significantly higher than for any other state. This peak corresponds to the state that Grover's algorithm has identified as the solution to the search problem.

As a search algorithm, Grover's algorithm works by repeatedly applying an oracle that inverts the phase of the solution state, and then a diffusion operator that amplifies the amplitude of the state(s) with a negative phase. The result is that the amplitude (and thus the probability upon measurement) of the correct state becomes much larger than that of the incorrect states.

d)

The original number of iterations was set to be optimal, which was calculated like this $\lfloor \frac{\pi}{4} * \sqrt{\frac{2^n}{m}} \rfloor$, where n is the number of qubits and m is the number of marked states. In this case, $n = 6$ and $m = 1$, so the number of iterations is $\frac{\pi}{4} * \sqrt{\frac{2^6}{1}} = 6.2832$, so it was taken as 6. First, I tried to run the circuit with 8 iterations, and the result is the following:

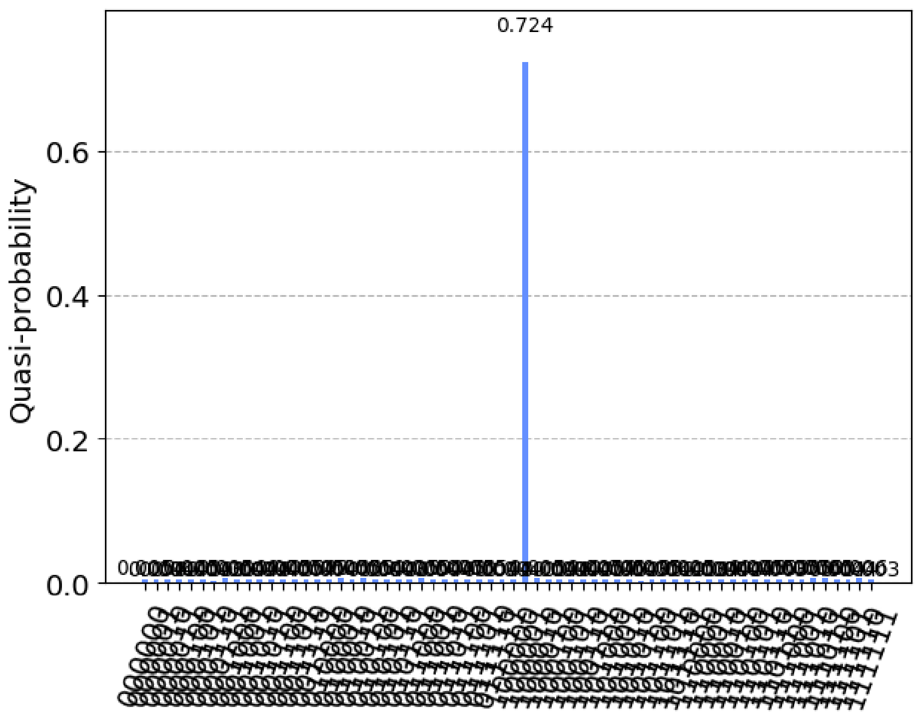

Figure 5: Plot of the distributions with 8 iterations

Although, there are many points on the x axis now, so it is checking some other states, and the peak point is lower than the optimal number of iterations, the result is clearly visible. The peak is still the same with the quasi-probability 0.724.

Then, I tried to run the circuit with 10 iterations, and the result is the following:

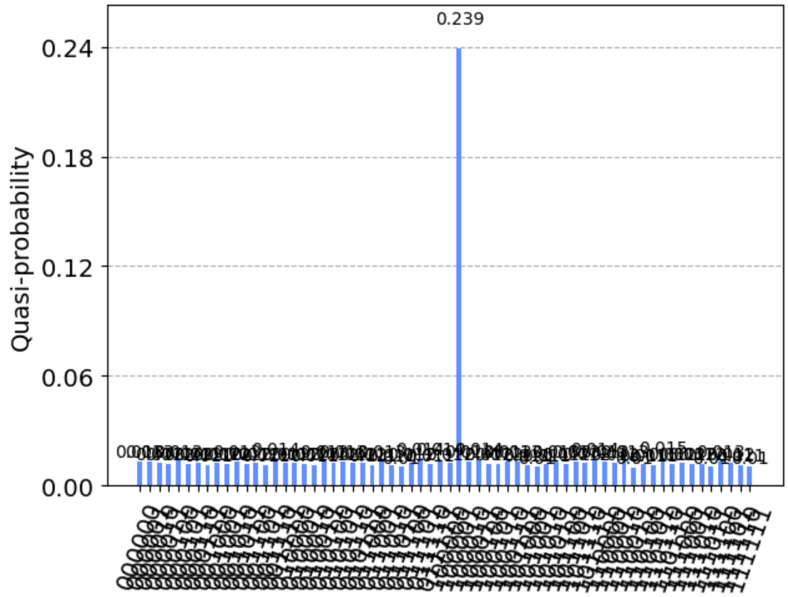

Figure 6: Plot of the distributions with 10 iterations

The peak is much more lower than the one with the optimal number of iterations, and the probability of it is 0.239. Besides, now other points on the x axis are have higher quasi-probabilities than the previous ones. However, they are still much lower than 0.239, so it is still working.

Finally, when I tried to run the circuit with 12 iterations, the result is the following:

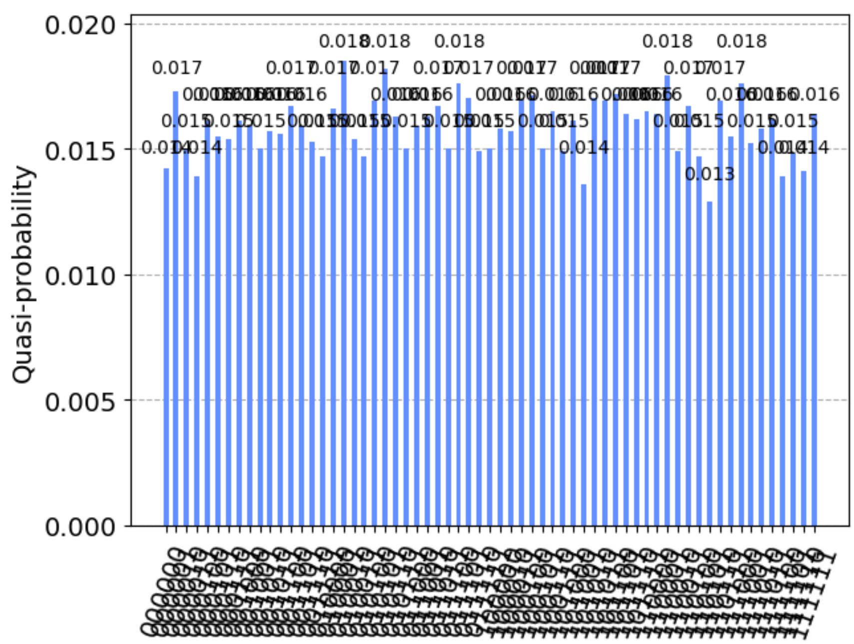

Figure 7: Plot of the distributions with 12 iterations

From this plot, nothing much can be understood, the probabilities are very close to each other and not even visible in the plot. Thus, the algorithm is not working with 12 iterations.

The reason why such a thing happened is that Grover's algorithm iteration is rotation of any state in the circle over an angle 2Θ where Θ is $arcsin(\frac{1}{\sqrt{n}})$, and the algorithm works on (2k+1)Θ where k is an integer. So, the algorithm works with 6 iterations, but not with 12 iterations. Also, I tried further, and it is starting to work again when I increased the number of iterations up to 18, which is the same as 6, then again until 24, the success rate is decreasing, and with 24, it is not working as the case for 12. This is sometimes referred to as "overshooting" the target state.

Part 2

a)

1. True

2. False. QFT circuit contains mixture of Hadamard and phase gates, but not CNOT gates.

3. False. The phase gate R_k used in QFT is $R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$.

4. True

5. False. Quantum phase estimation is a crucial part of Shor's factorization algorithm as it is used to find the order of an element which is key to factorization.

b)

Here is the circuit for the QFT with my id "100001" in binary:

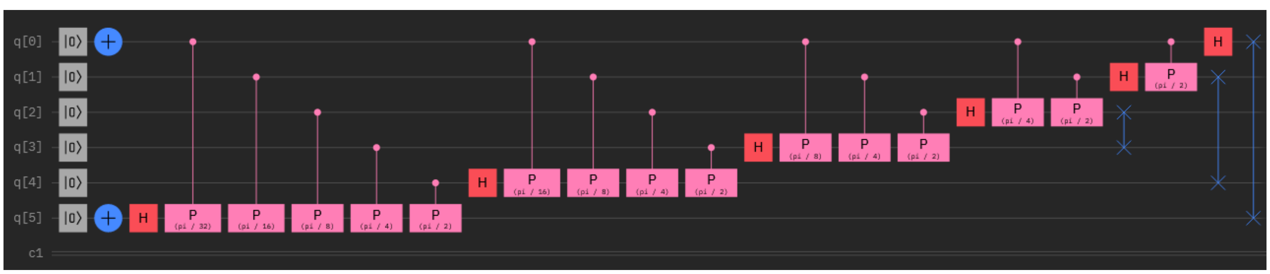

Figure 8: QFT circuit

Here is the plot of the results, not quite readable:

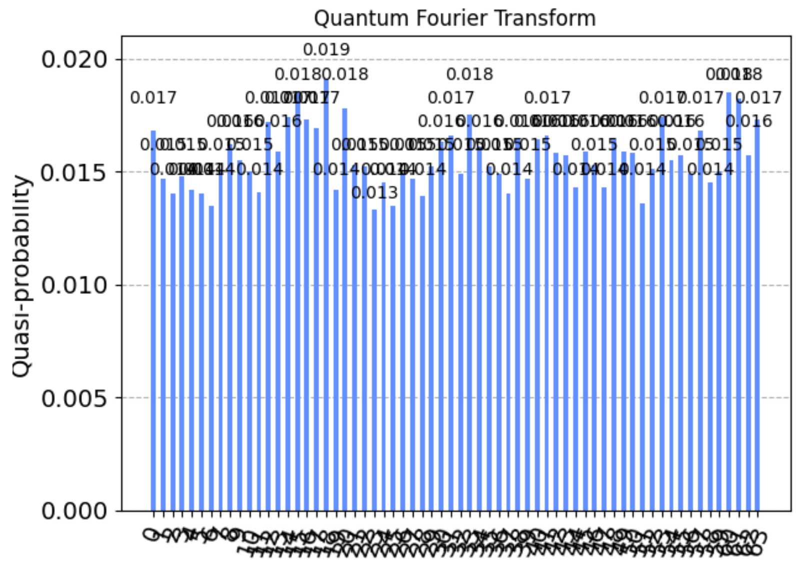

Figure 9: Plot of the results

As it is not readable, I printed the results as a table:

State	Prob	State	Prob	State	Prob	State	Prob
000000	0.0168	000001	0.0147	000010	0.0140	000011	0.0148
000100	0.0142	000101	0.0140	000110	0.0135	000111	0.0151
001000	0.0162	001001	0.0155	001010	0.0150	001011	0.0141
001100	0.0172	001101	0.0159	001110	0.0174	001111	0.0185
010000	0.0173	010001	0.0169	010010	0.0191	010011	0.0142
010100	0.0178	010101	0.0153	010110	0.0152	010111	0.0133
011000	0.0145	011001	0.0135	011010	0.0149	011011	0.0147
011100	0.0139	011101	0.0152	011110	0.0163	011111	0.0166
100000	0.0149	100001	0.0175	100010	0.0161	100011	0.0152
100100	0.0149	100101	0.0140	100110	0.0165	100111	0.0147
101000	0.0164	101001	0.0166	101010	0.0158	101011	0.0157
101100	0.0143	101101	0.0159	101110	0.0153	101111	0.0143
110000	0.0164	110001	0.0159	110010	0.0158	110011	0.0136
110100	0.0151	110101	0.0174	110110	0.0155	110111	0.0157
111000	0.0149	111001	0.0168	111010	0.0145	111011	0.0150
111100	0.0185	111101	0.0182	111110	0.0157	111111	0.0173

Figure 10: Table of the results

With these results, in summary, QFT has taken the inital state $|100001\rangle$, and transformed it into a complex superposition of all possible states, with amplitudes that reflect the Fourier transform of the initial state. This non-uniform distribution is a direct consequence of the input state and the QFT operation.