|  |  |
| --- | --- |
| **2 marks questions** | |
| 1 | State the difference between positive numbers and natural numbers.  **Natural Numbers:** The basic counting numbers starting from 1 (1, 2, 3, 4, ...).   * **Positive Numbers:** All numbers greater than zero, including natural numbers, fractions, and decimals (e.g., 1.5, 3/4, 100.25). |
| 2 | What do you understand by imaginary number?  **Imaginary Unit (i):** The imaginary unit is represented by 'i', where i² = -1. It's the foundation for imaginary numbers.   * **Imaginary Numbers:** Numbers expressed in the form of a real number multiplied by the imaginary unit (e.g., 2i, -5i, 3.14i). |
| 3 | What is the value of i3 ?  i^3 = i^2 \* i = (-1) \* i = -i |
| 4 | What is complex number?  **Complex Numbers:** Numbers of the form 'a + bi', where 'a' and 'b' are real numbers, and 'i' is the imaginary unit.   * **Example:** 3 + 4i |
| 5 | What do you understand by field?  **In mathematics, a field is a set of numbers where you can perform the following operations:**   * + Addition (+)   + Subtraction (-)   + Multiplication (×)   + Division (÷) - with the exception of division by zero |
| 6 | When a field can be called algebraically complete?  An algebraically complete field is a field where every polynomial equation with coefficients from that field has a root (or solution) within the same field.   * **Example:** The field of complex numbers is algebraically complete. |
| 7 | How you can change sign of a complex number?  **Changing the sign of a complex number**   * To change the sign of a complex number, multiply both the real and imaginary parts by -1. * **Example:** If z = 3 + 4i, then -z = -3 - 4i |
| 8 | What is conjugation?  **Complex conjugate**   * The complex conjugate of a complex number 'a + bi' is 'a - bi'. * **Notation:** The complex conjugate of z is often denoted as z̅ |
| 9 | What do you understand by complex conjugate?   * The complex conjugate of a complex number 'a + bi' is 'a - bi'. * **Notation:** The complex conjugate of z is often denoted as z̅ |
| 10 | Write a C^4 zero vectors. |
| 11 | What can be called as an inverse of a vector?   * **Additive Inverse:** In linear algebra, the inverse of a vector is more accurately called its "additive inverse." The additive inverse of a vector 'v' is the vector '-v', which has the same magnitude as 'v' but points in the opposite direction. When you add a vector and its additive inverse, you get the zero vector. |
| 12 | What is abelian group?   * **Abelian Group:** A group in which the order of operations does not matter. In other words, for any two elements 'a' and 'b' in the group, a \* b = b \* a (the commutative property). * **Examples:**   + The set of integers under addition.   + The set of non-zero real numbers under multiplication. |
| 13 | What is Hermitian matrix?   * **Hermitian Matrix:** A square matrix that is equal to its own conjugate transpose. In other words, the element at row 'i', column 'j' is equal to the complex conjugate of the element at row 'j', column 'i'. * **Properties:**   + The eigenvalues of a Hermitian matrix are always real.   + Hermitian matrices are important in quantum mechanics. |
| 14 | What is unitary matrix?   * **Unitary Matrix** A square matrix whose conjugate transpose is equal to its inverse. In other words, if U is a unitary matrix, then U\* U = I, where U\* is the conjugate transpose and I is the identity matrix. * **Properties:**   + Unitary matrices preserve lengths and angles.   + They are used in quantum mechanics to represent transformations. |
| 15 | What is bit?   * **Bit:** The fundamental unit of information in computing. It stands for "binary digit" and can have only two possible values: 0 or 1. |
| 16 | What is qubit?   * **Qubit:** The basic unit of information in quantum computing. Unlike a classical bit, a qubit can exist in a superposition of both the 0 and 1 states simultaneously. |
| 17 | What is qubyte?   * **Qubyte** A collection of eight qubits. |
| 18 | What are the basic logic gates?   * **AND Gate:** Outputs 1 only if both inputs are 1. * **OR Gate:** Outputs 1 if either (or both) inputs are 1. * **NOT Gate:** Inverts the input (0 becomes 1, and 1 becomes 0). |
| 19 | What are the universal logic gates?   * **NAND Gate:** (NOT-AND): A combination of the AND and NOT gate. It outputs 0 only if both inputs are 1. * **NOR Gate:** (NOT-OR): A combination of the OR and NOT gate. It outputs 1 only if both inputs are 0. * **Importance:** NAND and NOR gates are universal because any other logic gate can be constructed from a combination of them. |
| 20 | What is quantum gate?   * **Quantum Gate:** The basic building block for quantum circuits. They operate on qubits and perform unitary transformations upon them. * **Examples:**   + Hadamard Gate (H)   + Pauli Gates (X, Y, Z)   + CNOT Gate |
| 21 | What is encryption?   * **Encryption:** The process of transforming readable data (plaintext) into an unreadable form (ciphertext) to protect its confidentiality. Encryption uses an algorithm and a key to make the data unreadable to anyone without the corresponding decryption key. |
| 22 | What is decryption?   * **Decryption:** The process of reversing encryption. It involves using a key and an algorithm to convert the ciphertext back into its original, readable plaintext format. |
| 23 | What is ciphertext?   * **Ciphertext:** The unreadable, scrambled output of the encryption process. It conceals the original plaintext message to prevent unauthorized access. |
| 24 | What is eavesdropping?   * **Eavesdropping:** The act of secretly intercepting or listening to communications without authorization. This is a major threat in communication systems where data security is crucial. |
| 25 | What does the term superposition refer to?   * **Superposition (in quantum mechanics):** A fundamental principle of quantum mechanics where a quantum system (like a qubit) can exist in multiple states simultaneously until it is measured. For example, a qubit in superposition could have a probability of being both 0 and 1 at the same time. |
| 26 | How is a qubit in quantum computing different from a regular bit in  classical computing?   * **Classical Bit:** Can only be in one of two states: 0 or 1. * **Qubit:** Can exist in a superposition of both the 0 and 1 states simultaneously. This superposition allows for more complex computations and the potential for exponentially faster speeds in certain algorithms. |
| 27 | In quantum computing, what is the basic unit of information?   * **The qubit** is the basic unit of information in quantum computing. |
| 28 | Which part of the drug discovery life cycle can quantum computing impact the most?   * **Quantum computing is expected to have the biggest impact on the early stages of drug discovery:**   + **Molecular Simulation:** Quantum computers can more accurately simulate complex molecular interactions, aiding in the identification of potential drug candidates.   + **Protein Design:** Designing new protein structures for therapeutic targets.   + **Optimization:** Optimizing drug delivery routes and clinical trial designs. |
| 29 | What does it mean when we say that we are currently in the noisy,  intermediate scale quantum computing stage?   * **NISQ** refers to the current era of quantum computing where we have quantum computers with a limited number of qubits (around 50-100) and significant noise levels. Noise in quantum computers can cause errors in calculations. This stage is characterized by ongoing research to improve qubit quality and develop algorithms that can work within these limitations. |
| 30 | What is the purpose of “post-quantum” cryptography?   * **Post-quantum cryptography** is the development of cryptographic algorithms that are believed to be secure against attacks by quantum computers. This is important because many current encryption algorithms (like RSA) will be easily broken by powerful quantum computers in the future. |
| 31 | Why might businesses be interested in using quantum computers?   * **Speed:** Quantum computers have the potential to solve certain problems exponentially faster than classical computers. This could lead to breakthroughs in optimization, simulation, and machine learning tasks that were previously intractable. * **New Solutions:** Quantum computing algorithms might uncover solutions to problems that are difficult or impossible to solve with classical computing. This could lead to new discoveries in materials science, drug development, and more. * **Security:** Quantum computing could have applications in developing new, more secure communication and cryptography methods. * **Competitive Advantage:** Early adopters of quantum computing might gain a significant competitive advantage in industries relying heavily on computationally intensive tasks. |
| 32 | What is a current concern regarding the advancement of quantum  computing?   * **Scalability:** Building and maintaining large-scale, fault-tolerant quantum computers is extremely challenging. * **Cost:** Quantum computers are expensive and resource-intensive to develop and operate. * **Skills Gap:** There's a shortage of experts with the knowledge needed to harness quantum hardware and develop quantum algorithms. * **Security Threat:** The ability of future quantum computers to break existing encryption poses a security risk for sensitive data and communications. |
| 33 | Which technology will quantum computing impact most significantly?  It's hard to pinpoint a single technology, but here are areas likely to see significant impact:   * **Materials Science:** Designing new materials with enhanced properties. * **Drug Discovery:** Accelerating the identification and development of new drugs. * **Artificial Intelligence:** Developing new machine learning algorithms and improving existing ones. * **Cryptography:** Breaking existing encryption and the development of post-quantum cryptography. |
| 34 | How could quantum computing benefit the Financial Service Industry?   * **Risk Analysis:** More accurate modeling of complex financial systems for risk assessment. * **Portfolio Optimization:** Finding optimal investment strategies across a vast range of assets. * **Fraud Detection:** Developing algorithms that can spot unusual patterns or anomalies in large datasets faster. * **Algorithmic Trading:** Exploiting potential arbitrage opportunities with greater speed. |
| 35 | Which problem is more classical effectively solved using quantum  computing rather than classical computer?  **Factoring Large Integers:** At the heart of some encryption systems (Shor's Algorithm)   * **Searching Unsorted Databases:** Potentially offering a quadratic speed-up (Grover's Algorithm) * **Simulating Quantum Systems:** Essential for material design and chemistry research. |
| 36 | How you can define complex number as an ordered pair?  A complex number 'z = a + bi' can be represented as an ordered pair of real numbers: z = (a, b), where 'a' is the real part and 'b' is the imaginary part. |
| 37 | Let c = 1 + i. What is its polar representation?  **Magnitude (r):** r = |c| = sqrt(1^2 + 1^2) = sqrt(2)   * **Angle (theta):** theta = arctan(b/a) = arctan(1/1) = 45 degrees (or pi/4 radians) * **Polar Representation:** c = sqrt(2) \* (cos(45°) + i\*sin(45°)) |
| 38 | What is complex vector space?  A vector space where the scalars are complex numbers. The elements of a complex vector space can be vectors whose components are complex numbers. |
| 39 | What can be depicted by C^4?  C^4 represents the 4-dimensional vector space over the field of complex numbers. Vectors in this space would have four components, each of which is a complex number. |
| 40 | What can be called as an inverse of a vector?  The more accurate term is **additive inverse**. It's a vector that, when added to the original vector, results in the zero vector. For a vector 'v', its additive inverse is '-v'. |
| 41 | When a set of vectors can be called linearly independent?  A set of vectors is linearly independent if none of the vectors in the set can be expressed as a linear combination of the others. In other words, the only way to get the zero vector by combining the vectors with scalar coefficients is if all the coefficients are zero. |
| 42 | How a qubit pair can be written?  A qubit pair can be represented using tensor product notation. If you have two individual qubits |ψ₁⟩ and |ψ₂⟩, their combined state is written as:  |ψ₁⟩ ⊗ |ψ₂⟩ |
| 43 | How a two bit qubit can be expressed?  A two-qubit system has four possible basis states:   * |00⟩ * |01⟩ * |10⟩ * |11⟩ |
| 44 | What do you understand by probabilistic system?  A probabilistic system is a system where outcomes or events aren't deterministic but are instead governed by probabilities. The exact outcome of a single event might be unpredictable, but the overall distribution of possible outcomes can be described using probability theory. |
| 45 | A mathematician is trying to break down a large number into smaller prime numbers that, when multiplied, would arrive at that larger number. what is this process called?  This process is called **prime factorization**. It involves finding the prime numbers that, when multiplied together, give you the original number.   * **Example:** The prime factorization of 12 is 2 x 2 x 3. |
| 46 | What is the term for breaking a larger number apart into smaller numbers that can be multiplied together to get a specific result?  This process is called **prime factorization**. It involves finding the prime numbers that, when multiplied together, give you the original number.   * **Example:** The prime factorization of 12 is 2 x 2 x 3. |
| 47 | Which phenomena of quantum mechanics used in Quantum computing?  **Superposition:** Qubits can be in a combination of 0 and 1 states simultaneously.   * **Entanglement:** Qubits can be linked so their states are correlated, even when separated. * **Interference:** Quantum computations can use interference patterns to manipulate probabilities and converge on solutions faster. |
| 48 | What can be states of a qubit?  **Basis States:** A qubit has two basis states: |0⟩ and |1⟩.   * **Superposition:** A qubit can be in a superposition of the basis states: α|0⟩ + β|1⟩ (where α and β are complex coefficients that satisfy |α|^2 + |β|^2 = 1). |
| 49 | In a Quantum circuit, If memory consists of n bits of information, then what will be the possible no. of states?  A quantum circuit with n qubits can be in a superposition of 2^n possible states. |
| 50 | When the information is between 0 and 1 in a quantum computer, what do we call this?  This refers to the probabilistic nature of a qubit in superposition. Before measurement, the qubit has a probability of collapsing to either the 0 or 1 state. This probability lies between 0 and 1, representing the degree of confidence for each possible outcome. |
| 51 | What does 'entanglement' mean?  **Entanglement:** A unique quantum phenomenon where two or more particles become inextricably linked. The state of each particle cannot be described independently—measuring one instantly affects the state of the other(s), even if they are physically separated by large distances. |
| 52 | Quantum encryption keys can be hacked and broken into. True or False?  **Mostly False (with nuances):** Quantum key distribution (QKD), a form of quantum cryptography, is theoretically very secure due to the principles of quantum mechanics. However:   * + **Implementation Flaws:** Errors or vulnerabilities in the implementation of QKD systems could potentially be exploited.   + **Side-Channel Attacks:** Attacks targeting the physical devices rather than the underlying principles might be possible. |
| 53 | Why are we not able to use quantum computers in our homes?  **Size and Complexity:** Current quantum computers are large and delicate machines, often requiring highly controlled environments like cryogenic temperatures.   * **Cost:** Building and maintaining quantum computers is very expensive with the technology still in its early stages. * **Specialized Use:** Quantum computers excel at specific problem types. They are not intended to replace classical computers for everyday tasks. |
| 54 | What is supercomputer?  **Supercomputer:** An extremely powerful computer with far more processing power and memory than standard computers. They are used for computationally intensive tasks like scientific simulations, weather forecasting, and large-scale data analysis. |
| 55 | How quantum computer differs from super computer?  **Fundamental Approach:** Supercomputers utilize classical computing principles, relying on bits. Quantum computers harness quantum phenomena like superposition and entanglement for entirely different kinds of computation.   * **Problem-Solving:** Supercomputers excel at tasks that can be broken into parallelizable operations. Quantum computers are better suited for problems with exponential complexity, like factoring large numbers or simulating quantum systems. |
| 56 | Convert 23D916 to binary, decimal and octal.  **Binary:** 10001111011001   * **Decimal:** 145369 * **Octal:** 47551 |
| 57 | Solve for the decimal equivalent of (101101.10101)2  **Decimal:** 45.65625 |
| 58 | Write De-Morgan’s law and verify it with the truth table.  **First Law:** The complement of the OR of two variables is equal to the AND of their complements. ¬(A ∨ B) = ¬A ∧ ¬B   * **Second Law:** The complement of the AND of two variables is equal to the OR of their complements. ¬(A ∧ B) = ¬A ∨ ¬B |
| 59 | Which possible Attacks in Quantum Cryptography can take place?  **Truth table verification:** You can create a truth table with columns for A, B, the expressions on both sides of the equation, and compare their results. The results will match, verifying the laws. |
| 60 | Which two principles of quantum mechanics on which quantum  cryptography are dependent?  **No-Cloning Theorem:** It's impossible to create a perfect copy of an unknown quantum state. This prevents eavesdropping without being detected.   * **Quantum Measurement:** Measuring a quantum state alters it. This ensures that any eavesdropping attempt will disturb the transmitted qubits and be detectable. |
| 61 | What is the greatest threat to public cryptography or asymmetric  algorithms?  While quantum computers pose a significant threat in the future, some current concerns regarding public cryptography (asymmetric algorithms) include:   * Poor key management practices: If public or private keys are weak, improperly stored, or leaked, it can compromise the entire cryptographic system. * Weaknesses in specific algorithms or implementations: Not all asymmetric algorithms are created equal. Some may have inherent vulnerabilities or weaknesses that could be exploited by attackers. |
| 62 | What is active attack?  Active Attack: Attempts to modify messages, delete them, or prevent them from being delivered. |
| 63 | What is passive attack?  Passive Attack: Aims to eavesdrop on communication to gather information without altering the messages. |
| 64 | What is the difference between symmetric and asymmetric cryptography?  I |
| 65 | What is information gain?  Information gain is a concept from information theory that measures the reduction in uncertainty about a message after gaining information. It's not directly calculated here but plays a role in analyzing the effectiveness of cryptographic systems. |
| 66 | What is the modulus of c = (1-i)?  You can calculate the modulus (absolute value) of a complex number 'c' using the following formula:  |c| = sqrt(real^2 + imag^2)  Here, 'real' is the real part of the complex number and 'imag' is the imaginary part.  Modulus of c1 (1 - i): sqrt(1^2 + (-1)^2) = 1.41421 (approx) |
| 67 | Calculate the modulus of c = (4-3i)  Modulus of c2 (4 - 3i): sqrt(4^2 + (-3)^2) = 5.0 |
| 68 | Consider the complex numbers c = 3 + 4i. Find the length of the vector.  The modulus (absolute value) of a complex number also represents its length as a vector. Therefore, the lengths of the complex numbers you provided are the same as their moduli that we calculated previously. |
| 69 | What is the meaning of AND |11>? |
| 70 | Why AND gate is not reversible?  The AND gate is not reversible because its output (0) can be produced by multiple input combinations. For example, both (0, 0) and (0, 1) will result in an output of 0. Without knowing the original inputs, it's impossible to determine the exact state from the output alone. This makes it unsuitable for scenarios where you need to recover the original data from the output. |
| 71 | Let (x+3) + i(y-2) = (5+2i), find x and y.  Here's how to solve for x and y in the complex number equation:  **1. Separate Real and Imaginary Parts:**  A complex number equation is considered equal if both the real and imaginary components are equal. So, we can set up two separate equations:   * **Real Parts:** x + 3 = 5 * **Imaginary Parts:** y - 2 = 2   **2. Solve for x and y:**   * Solving for x: x + 3 = 5 x = 5 - 3 x = 2 * Solving for y: y - 2 = 2 y = 2 + 2 y = 4   **Therefore, x = 2 and y = 4** |
| 72 | Calculate the value of  √(-16) = √(16 \* -1)   * + √(-16) = √(16) \* √(-1)   + √(-16) = 4i   **Therefore, the square root of -16 is 4i.** |
| 73 | Calculate the value of  Here's how to simplify the expression:  **1. Imaginary Unit (i):** Recall that the imaginary unit 'i' represents √(-1)  **2. Break Down the Square Roots:**   * √(-25) = √(25 \* -1) = 5i * √(-4) = √(4 \* -1) = 2i * √(-9) = √(9 \* -1) = 3i   **3. Substitute and Simplify:**  sqrt(-25) + 3 \* sqrt(-4) + 2 \* sqrt(-9) = 5i + (3 \* 2i) + (2 \* 3i) = 5i + 6i + 6i = 17i  **Therefore, sqrt(-25) + 3 \* sqrt(-4) + 2 \* sqrt(-9) = 17i** |
| 74 | What is the probability of getting a sum as 3 if a dice is thrown?  Possible outcomes: 6 (sides of a die)   * Favorable outcomes: 2 (1+2, 2+1) * Probability: 2/6 = 1/3 |
| 75 | In a box, there are 8 orange, 7 white, and 6 blue balls. If a ball is picked up randomly, what is the probability that it is neither orange nor blue?  Total balls: 8 + 7 + 6 = 21   * White balls: 7 * Probability: 7/21 = 1/3 |
| 76 | If a number is selected at random from the first 50 natural numbers, what will be the probability that the selected number is a multiple of 3 and 4?  Total numbers: 50   * Multiples of 12 within 50: 4 (12, 24, 36, 48) * Probability: 4/50 = 2/25 |
| 77 | What are the limitations of present quantum computers?  **Noise:** Quantum systems are very sensitive to environmental disturbances. Noise can cause errors in calculations.   * **Scalability:** Building quantum computers with large numbers of stable qubits is difficult and expensive. * **Limited algorithms:** The set of quantum algorithms that offer significant advantages over classical algorithms is still relatively small. |
| 78 | What is the role of error correction when working with quantum  computers?  **Crucial for Reliability:** Quantum error correction (QEC) is vital because noise is a major obstacle. QEC techniques use redundancy and encoding to detect and correct errors, allowing for more reliable computations.   * **Challenges:** Developing effective QEC methods is complex and adds resource overhead for quantum computers, further hindering scalability. |
| 79 | What is the relationship between quantum computing and machine  learning?  **Potential Speed-ups:** Quantum computing could accelerate certain machine learning tasks, particularly linear algebra operations used in training models.   * **Quantum Machine Learning Algorithms:** Researchers are exploring algorithms that leverage quantum phenomena for tasks like classification and pattern recognition. * **Early Stages:** The field is still nascent, and it's unclear which real-world machine learning applications will see the most significant benefits. |
| 80 | Is a qubit still considered digital?  While a qubit can ultimately be measured as a 0 or 1 (like a classical bit), it's more accurate to say that a qubit has analog qualities due to the following:   * **Superposition:** Before measurement, a qubit exists in a superposition of both 0 and 1 states, represented by complex coefficients. This is fundamentally different from the discrete states of a classical bit. * **Analog Nature:** The probabilities associated with a qubit's state are continuous values, not simply on or off. |
|  |  |
| **5 marks Question** | |
| 1 | Let, c1 = (3 - i) and c2 = (1+4i) and Compute c1 + c2 and c1 × c2. |
| 2 | Let, c1 = (-3 + i) and c2 = (2 - 4i) and Compute c1 + c2 and c1 × c2. |
| 3 | Verify that the complex number (-1+i) is a solution for the polynomial equation |
| 4 | Write a program (any language) that accepts two complex numbers and outputs their sum and their product. |
| 5 | Let c1 =(3, -2) and c2 = (1,2). Multiply them.  **Understanding the Notation**  When complex numbers are written as ordered pairs like (a, b), this represents a + bi:   * The first element ('a') is the real part of the complex number. * The second element ('b') is the imaginary part of the complex number.   **Multiplication Steps**   1. **Use the Distributive Property (FOIL):** (3, -2) \* (1, 2) = (3 \* 1, 3 \* 2, -2 \* 1, -2 \* 2i) 2. **Simplify Each Term:** (3, 6, -2, -4i) 3. **Combine Like Terms and Use i² = -1** (3 - 2) + (6 - 4)i = 1 + 2i   **Therefore, the product of (3, -2) and (1, 2) is 1 + 2i.** |
| 6 | Let c1 =(-3, -1) and c2 = (1,-2). Calculate the product.  **Understanding the Notation:**  The ordered pair notation (a, b) represents a complex number in the form a + bi.  **Steps for Multiplication:**   1. **Distributive Property (FOIL):** (-3, -1) \* (1, -2) = (-3 \* 1, -3 \* -2, -1 \* 1, -1 \* -2i) 2. **Simplify Each Term:** (-3, 6, -1, 2i) 3. **Combine Like Terms and Use i² = -1** (-3 - 1) + (6 + 2)i = -4 + 8i   **Therefore, the product of c1 and c2 is -4 + 8i.** |
| 7 | Let, c1 = (-2+i) and c2 = (1+2i). Compute c1/c2.  **Method: Complex Conjugates**   1. **Find the Conjugate of the Denominator (c2):** The conjugate of (1 + 2i) is (1 - 2i). 2. **Multiply Both Numerator and Denominator by the Conjugate:**   [(-2 + i) \* (1 - 2i)] / [(1 + 2i) \* (1 - 2i)]   1. **Expand the Products (use FOIL):**   (-2 + 4i + i - 2i^2) / (1 - 4i^2)   1. **Simplify Using i² = -1**   (0 + 5i) / (5)   1. **Separate Real and Imaginary Components**   0/5 + (5i)/5 = 0 + i  **Therefore, c1 / c2 = 0 + i (or simply 'i')** |
| 8 | Let, c1 = (3i) and c2 = (-1 - i). Compute c1/c2.  **Method: Complex Conjugates**   1. **Conjugate of the Denominator (c2):** The conjugate of (-1 - i) is (-1 + i). 2. **Multiply Both Numerator and Denominator by the Conjugate:**   [(3i) \* (-1 + i)] / [(-1 - i) \* (-1 + i)]   1. **Expand the Products (use FOIL):**   (-3i + 3i^2) / (1 - i^2)   1. **Simplify Using i² = -1**   (-3i - 3) / (2)   1. **Separate Real and Imaginary Components**   -3/2 - (3/2)i  **Therefore, c1 / c2 = -3/2 - (3/2)i** |
| 9 | Show that conjugation respects addition, i.e., c1 + c2 = c2 + c1. |
| 10 | Show that conjugation respects multiplication, i.e., c1 × c2 = c1 × c2 |
| 11 | Describe complex plane with proper diagram. |
| 12 | Verify that multiplication by (-1,0) changes the sign of the real and imaginary components of a complex number. |
| 13 | Discuss parallelogram rule for addition of vectors. |
| 14 | Discuss parallelogram rule for subtraction of vectors.  **Alternatively: Triangle Rule Interpretation**  You can also visualize vector subtraction using the triangle rule:   1. Place the tail of vector B at the tip of vector A. 2. The vector drawn from the tail of A to the tip of B represents A - B.   **Example**  [Image of parallelograms demonstrating vector subtraction](https://study.com/skill/learn/how-to-add-or-subtract-two-vectors-by-using-the-parallelogram-method-explanation.html)study.com  [Opens in a new window](https://study.com/skill/learn/how-to-add-or-subtract-two-vectors-by-using-the-parallelogram-method-explanation.html)  parallelograms demonstrating vector subtraction  Consider:   * Vector A: 4 units, pointing North. * Vector B: 2 units, pointing East.   To find A - B:   1. Draw vector A as a vertical line. 2. Find -B: A vector of 2 units pointing West. 3. Place the tail of -B at the tip of A. 4. Complete the parallelogram. The diagonal starting from the tail of A to the opposite corner represents the result A - B. |
| 15 | Write a program (any language) that converts a complex number from its Cartesian representation to its polar representation and vice versa.  import math  def cartesian\_to\_polar(real, imag):  """Converts a complex number from Cartesian (rectangular) form to polar form.  Args:  real: The real part of the complex number.  imag: The imaginary part of the complex number.  Returns:  A tuple containing the magnitude (radius) and angle (theta) in degrees.  """  magnitude = math.sqrt(real\*\*2 + imag\*\*2)  angle = math.degrees(math.atan2(imag, real)) # Angle in degrees  return magnitude, angle  def polar\_to\_cartesian(magnitude, angle):  """Converts a complex number from polar form to Cartesian (rectangular) form.  Args:  magnitude: The magnitude (radius) of the complex number.  angle: The angle (theta) in degrees of the complex number.  Returns:  A tuple containing the real and imaginary parts of the complex number.  """  radians = math.radians(angle) # Convert angle to radians  real = magnitude \* math.cos(radians)  imag = magnitude \* math.sin(radians)  return real, imag  # Example usage  # Cartesian to polar  real\_part = 3  imag\_part = 4  magnitude, angle = cartesian\_to\_polar(real\_part, imag\_part)  print("Polar representation: magnitude = {:.2f}, angle = {:.2f} degrees".format(magnitude, angle))  # Polar to Cartesian  magnitude = 2  angle = 60  real\_part, imag\_part = polar\_to\_cartesian(magnitude, angle)  print("Cartesian representation: real = {:.2f}, imaginary = {:.2f}".format(real\_part, imag\_part)) |
| 16 | Draw the complex numbers c1 = (2-i) and c2 = (2+i) in the complex plane, and subtract them using the parallelogram rule. Verify that you would get the same result as subtracting them algebraically. |
| 17 | Take any C^5 vector and take another C^5 vector and prove that second one is an inverse of a first one. |
| 18 | Add the given vectors    **Steps:**   1. **Identify Corresponding Components:**    * V = [6 - 4i, 7 + 3i, 4.2 - 8.1i, -3i]    * W = [16 + 2.3i, -7i, 6, -4i] 2. **Add Real Components:**    * 6 + 16 = 22    * 7 + 0 = 7 (Note: The second vector, W, doesn't have a real part in the second position)    * 4.2 + 6 = 10.2 3. **Add Imaginary Components:**    * -4 + 2.3 = -1.7    * 3 - 7 = -4    * -8.1 + 0 = -8.1    * -3 - 4 = -7 4. **Combine Results:**    * Resulting Vector: [22 - 1.7i, 7 - 4i, 10.2 - 8.1i, -7i ]   **Therefore, the sum of vectors V and W is [22 - 1.7i, 7 - 4i, 10.2 - 8.1i, -7i].** |
| 19 | What can be possible eigen values for the given matrix?      Here's how to find the possible eigenvalues for the given matrix:  **1. Characteristic Equation**  The characteristic equation is found by taking the determinant of (A - λI) and setting it to zero:  det(A - λI) = 0  For the matrix A = [{-6, 3}, {4, 5}], this expands to:  det ([{-6, 3}, {4, 5}] - λ \* [[1, 0], [0, 1]]) = 0  det([{-6 - λ, 3}, {4, 5 - λ}]) = 0  **2. Calculate the Determinant**  (-6 - λ)(5 - λ) - (3)(4) = 0  **3. Solve the Quadratic Equation**  Expand and simplify:  -30 + λ + λ^2 - 12 = 0  λ^2 + λ - 42 = 0  Factor the equation:  (λ + 7)(λ - 6) = 0  **4. Eigenvalues**  The solutions to this equation are the eigenvalues:   * λ₁ = -7 * λ₂ = 6   **Therefore, the possible eigenvalues for the given matrix are -7 and 6.** |
| 20 | Find eigen vector for the given matrix with a eigen  value of 7.      Absolutely! Let's find the eigenvector corresponding to the eigenvalue λ = 7 for the matrix A = [{-6, 3}, {4, 5}].  **Recall the Eigenvector Equation**  An eigenvector (v) corresponding to an eigenvalue (λ) satisfies the following equation:  (A - λI) v = 0  **Steps to Find the Eigenvector**   1. **Substitute Values:**    * A = [{-6, 3}, {4, 5}]    * λ = 7    * I (Identity Matrix) = [[1, 0], [0, 1]] 2. **Calculate (A - λI):**   [{-6, 3}, {4, 5}] - 7 \* [[1, 0], [0, 1]] = [{-13, 3}, {4, -2}]   1. **Solve the System of Equations:**   We need to solve the following system of equations to find the components x and y of eigenvector v = [x, y]:  -13x + 3y = 0  4x - 2y = 0   1. **Solution (Approach)**    * **Multiple Solutions:** This system has infinitely many solutions because the equations are dependent. We can express the solution in terms of a free variable.    * **Finding one Solution:** Let's choose y = 13. Substituting into the first equation, we get x = 3.   **Eigenvector**  One possible eigenvector corresponding to the eigenvalue λ = 7 is [3, 13]. Note that any non-zero scalar multiple of this vector is also an eigenvector. |
| 21 | Find eigen vector for the given matrix with a eigen  value of 6. |
| 22 | When following set of vectors can be called as linearly independent?    **Determinant:** If the determinant of the matrix formed by the vectors is non-zero, the vectors are linearly independent.   * + Form a matrix where each row is one of your vectors:   [ 1 1 1 ]  [ 0 1 1 ]  [ 0 0 1 ]   * + Calculate the determinant of this matrix. If it's non-zero, the vectors are linearly independent. |
| 23 | When following set of vectors can be called as linearly independent? |
| 24 | Show that the following set of vectors are not linearly independent. |
| 25 | Show that given matrix is Hermitian matrix. |
| 26 | Express inputs and output of NAND gate as matrix. |
| 27 | Show that NAND is a combination of NOT and AND. |
| 28 | Express inputs and output of NOR gate as matrix. |
| 29 | Explain controlled-NOT gate.  The controlled-NOT gate, also known as CNOT gate, is a fundamental quantum logic gate that operates on two qubits. Here's a breakdown of its functionality:  **Basic Idea**   * A CNOT gate uses one qubit as a "control" qubit and another as a "target" qubit. * The gate flips the target qubit (performs a NOT operation) only if the control qubit is in the state |1⟩ (one). * If the control qubit is in the state |0⟩ (zero), the target qubit remains unchanged.   **Qubit Representation**   * Qubits in quantum computing can be in a state of |0⟩, |1⟩, or a superposition of both, represented as a complex number.   **Symbol and Truth Table**   * The CNOT gate is often symbolized by a rectangle with a control line entering on top and two output lines, one for each qubit. A dot on the control line indicates the controlled operation.     **Applications**   * CNOT gates are essential building blocks for many quantum algorithms. * They are used in quantum teleportation, entanglement generation, and other quantum information processing tasks. * By combining CNOT gates with single-qubit rotations, any quantum circuit can be built in theory.   **Example**  Imagine a CNOT gate with:   * Control qubit: |1⟩ (one) * Target qubit: |0⟩ (zero)   After applying the CNOT gate, the target qubit becomes |1⟩, while the control qubit remains |1⟩.  **Key Points**   * CNOT gates enable conditional operations on qubits based on the state of another qubit. * They are crucial for manipulating and processing quantum information. * Understanding CNOT gates is a foundation for exploring the vast potential of quantum computing.   Do you have any questions about the controlled-NOT gate or its applications in quantum circuits? |
| 30 | Discuss pauli matrices.  Absolutely! Pauli matrices are incredibly important in quantum mechanics and quantum computing. Here's a discussion covering their key aspects:  **What are Pauli Matrices?**  The Pauli matrices are a set of three 2x2 complex Hermitian and unitary matrices commonly denoted by the Greek letter sigma (σ):   * **σ₁ (Pauli-X):**   σ₁ = [ 0 1 ]  [ 1 0 ]   * **σ₂ (Pauli-Y):**   σ₂ = [ 0 -i ]  [ i 0 ]   * **σ₃ (Pauli-Z):**   σ₃ = [ 1 0 ]  [ 0 -1 ]  **Key Properties**   * **Hermitian:** Each Pauli matrix is equal to its own conjugate transpose. This means they have real eigenvalues, which is important in quantum mechanics where observables are represented by Hermitian operators. * **Unitary:** Each Pauli matrix multiplied by its conjugate transpose results in the identity matrix. This means that Pauli gates in quantum computing are reversible. * **Involution:** Squaring any Pauli matrix results in the identity matrix (I): σ₁² = σ₂² = σ₃² = I * **Commutation and Anticommutation:** Pauli matrices have specific relationships with each other:   + Anticommutation: σ₁σ₂ + σ₂σ₁ = 0   + Commutation: σ₁σ₂ - σ₂σ₁ = 2i σ₃ (and cyclic permutations)   **Significance in Quantum Mechanics**   * **Spin Operators:** The Pauli matrices are closely connected to the representation of spin-1/2 particles (like electrons). Components of the spin angular momentum operator are expressed in terms of the Pauli matrices. * **Measurement:** The eigenvalues of Pauli matrices (+1 and -1) correspond to possible measurement outcomes for spin components along their respective axes.   **Quantum Computing Applications**   * **Single-Qubit Gates:** The Pauli matrices (X, Y, Z) form the basis for important single-qubit quantum gates, crucial for manipulating the state of qubits in quantum algorithms. * **Rotations:** Pauli gates represent rotations of a qubit's state on the Bloch sphere, a geometrical representation of a qubit's state space.   **Let me know if you'd like to delve deeper into any specific aspect of Pauli Matrices or explore their applications further!** |
| 31 | Explain square root of NOT gate.  The square root of NOT gate (√NOT) is a fundamental gate in quantum computing that, when applied twice in succession, has the same effect as the standard NOT gate. Let's break it down:  **The NOT Gate (X Gate)**   * The NOT gate flips the state of a qubit. * |0⟩ becomes |1⟩, and |1⟩ becomes |0⟩ * It's represented by the Pauli-X matrix.   **Understanding "Square Root" in Quantum Computing**   * **Analogy:** Think of the square root of 4 being 2, since 2 \* 2 = 4. In quantum gates, finding a "square root" involves finding a gate that, when applied twice, gives you the original gate. * **Unitary Property:** Quantum gates are represented by unitary matrices. Unitary matrices have the property that their inverse is equal to their conjugate transpose. This is important because quantum operations must be reversible.   **The √NOT Gate**  The square root of the NOT gate is represented by the following matrix:  √NOT = 1/2 \* [1 + i 1 - i]  [1 - i 1 + i]   * **Effect:** Unlike the NOT gate, √NOT doesn't simply flip qubit states. It creates a more complex superposition and rotation of the qubit's state on the Bloch sphere (a visual representation of a qubit's state). * **Check:** Applying √NOT twice (√NOT \* √NOT) results in the same effect as the NOT gate.   **Applications of √NOT**   * **Building Other Gates:** The √NOT gate is essential for constructing more complex quantum gates like the Hadamard gate (H) and phase gates (S, T). * **Quantum Algorithms:** √NOT finds application in various quantum algorithms where controlled rotations of qubit states are needed.   **Key Points**   * The √NOT gate provides more fine-grained control over qubit states than the standard NOT gate. * Understanding the √NOT gate unlocks a deeper understanding of how quantum gates manipulate quantum states. |
| 32 | Explain Deutsch gate.    **which is very similar to the Toffoli gate. If the inputs |x> and |y> are both |1>, then the phase shift operation R(θ) will act on the |z> input. Otherwise, the |z> will simply be the same as the |z>. When θ is not a rational multiple of π, D(θ) by itself is a universal three-qubit quantum gate. In other words, D(θ) will be able to mimic every other quantum gate.** |
| 33 | Find the probability of the qubit belonging to state |0> and |1>. |
| 34 | Calculate (i) 244,443 Mod 247 (ii) 18,154 Mod 247 (iii) 226,006 Mod 247. |
| 35 | Show that (i) 1,977 1 Mod 247 (ii) 16,183 15,442 Mod 247 (iii) 2,439,593 238,082 Mod 247 |
| 36 | If state of a classical deterministic system is expressed as X = [6, 2, 1, 5, 3, 10]^T, what we understand from this? |
| 37 | If state of a classical deterministic system is expressed as X = [5, 5, 0, 2, 0, 15]^T, what we understand from this? |
| 38 | In a probabilistic system a state is expressed by X = [1/ 5 , 3/10 , 1/2 ]^T, explain X.  In a probabilistic system, the vector X = [1/5, 3/10, 1/2]^T represents the **probability distribution** of the system being in each of its possible states.  Here's a breakdown of what the vector tells us:   * **Number of States:** The length of the vector (3 in this case) indicates the number of possible states the system can be in. * **Probability Values:** Each element in the vector represents the probability of the system being in the corresponding state.   + The first element (1/5) signifies a probability of 1 in 5 that the system is in the first state.   + The second element (3/10) signifies a probability of 3 in 10 that the system is in the second state.   + The third element (1/2) signifies a probability of 1 in 2 that the system is in the third state.   **Key Points:**   * The sum of all the elements in the vector must always equal 1. This ensures that all possible states are accounted for, and the probabilities add up to 100%. (In this case, 1/5 + 3/10 + 1/2 = 1, which validates the distribution). * This representation is commonly used in various probabilistic models like Markov chains, hidden Markov models, and Bayesian networks. |
| 39 | Normalize the vector [2 3i, 1 + 2i]^ T.  **Normalization Process**   1. **Calculate the Magnitude:** Find the magnitude (length) of the vector. The magnitude of a complex vector is calculated as:   ||v|| = sqrt(|a|^2 + |b|^2), where a and b are the components of the vector.  In this case: ||v|| = sqrt(|2 + 3i|^2 + |1 + 2i|^2) = sqrt(13 + 5) = sqrt(18)   1. **Divide by the Magnitude:** Divide each element of the vector by the calculated magnitude.   Normalized vector = [ (2 + 3i) / sqrt(18), (1 + 2i) / sqrt(18) ] |
| 40 | Find the unitary adjacency matrix for the given graph. |

**10 marks**

1)Verify that the multiplication of complex numbers is associative.

**Proof of Associativity**

Let's consider three complex numbers:

* z1 = a + bi
* z2 = c + di
* z3 = e + fi

We need to show: (z1 \* z2) \* z3 = z1 \* (z2 \* z3)

**Left-hand side:**

(z1 \* z2) \* z3

= [(a + bi) \* (c + di)] \* (e + fi)

= [(ac - bd) + (ad + bc)i] \* (e + fi)

= [(ac - bd)e - (ad + bc)f] + [(ac - bd)f + (ad + bc)e]i

**Right-hand side:**

z1 \* (z2 \* z3)

= (a + bi) \* [(c + di) \* (e + fi)]

= (a + bi) \* [(ce - df) + (cf + de)i]

= [(ace - bdf) - (acf + bde)] + [(acf + bde) + (ace - bdf)]i

**Comparing both sides:**

We can see that the real and imaginary components of the left-hand side and right-hand side expressions are identical.

**Conclusion**

Since (z1 \* z2) \* z3 = z1 \* (z2 \* z3) for arbitrary complex numbers z1, z2, and z3, we have verified that the multiplication of complex numbers is associative.

2)Derive the general equation for subtraction and division of complex numbers.

Absolutely! Here's the general derivation for subtraction and division of complex numbers:

**Subtraction of Complex Numbers**

Consider two complex numbers:

* z₁ = a₁ + b₁i
* z₂ = a₂ + b₂i

where a₁, a₂, b₁, and b₂ are real numbers, and 'i' is the imaginary unit.

**Subtraction Formula:**

The difference of two complex numbers is found by subtracting their real and imaginary parts separately:

z₁ - z₂ = (a₁ - a₂) + (b₁ - b₂)i

**Example:**

Let z₁ = 2 + 5i and z₂ = 4 - 2i

Then, z₁ - z₂ = (2 - 4) + (5 + 2)i = -2 + 7i

**Division of Complex Numbers**

Dividing complex numbers uses the concept of the complex conjugate.

**Complex Conjugate:**

The complex conjugate of z = a + bi is denoted by z\* and is given by:

z\* = a - bi

**Division Formula:**

To divide z₁ by z₂ (where z₂ ≠ 0), we proceed as follows:

* Multiply both numerator and denominator by the complex conjugate of the denominator (z₂\*).
* Expand the expressions in both numerator and denominator.
* Use the fact that i² = -1 to simplify.

(z₁ / z₂) = (z₁ \* z₂\*) / (z₂ \* z₂\*)

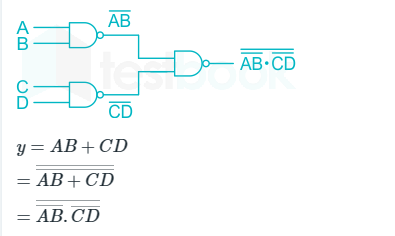
**Example:**

Let z₁ = 2 + 5i and z₂ = 4 - 2i

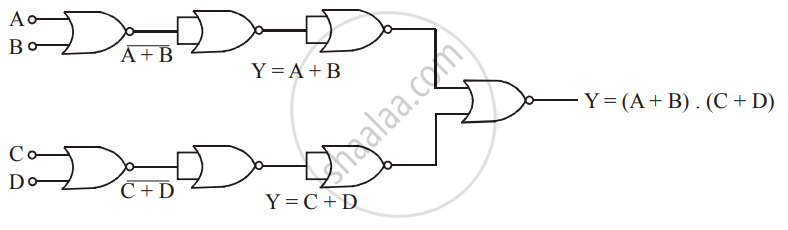
* Complex conjugate of z₂: z₂\* = 4 + 2i
* Division: (2 + 5i) / (4 - 2i) = ((2 + 5i) \* (4 + 2i)) / ((4 - 2i) \* (4 + 2i))
* Expand and simplify: = (8 + 18i + 10i²) / (16 + 4) = (-2 + 18i) / 20 = -1/10 + 9/10 i

Let me know if you would like to see any more worked examples!

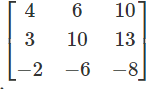
3)Illustrate the following expression: AB + CD, Using only NAND gate.

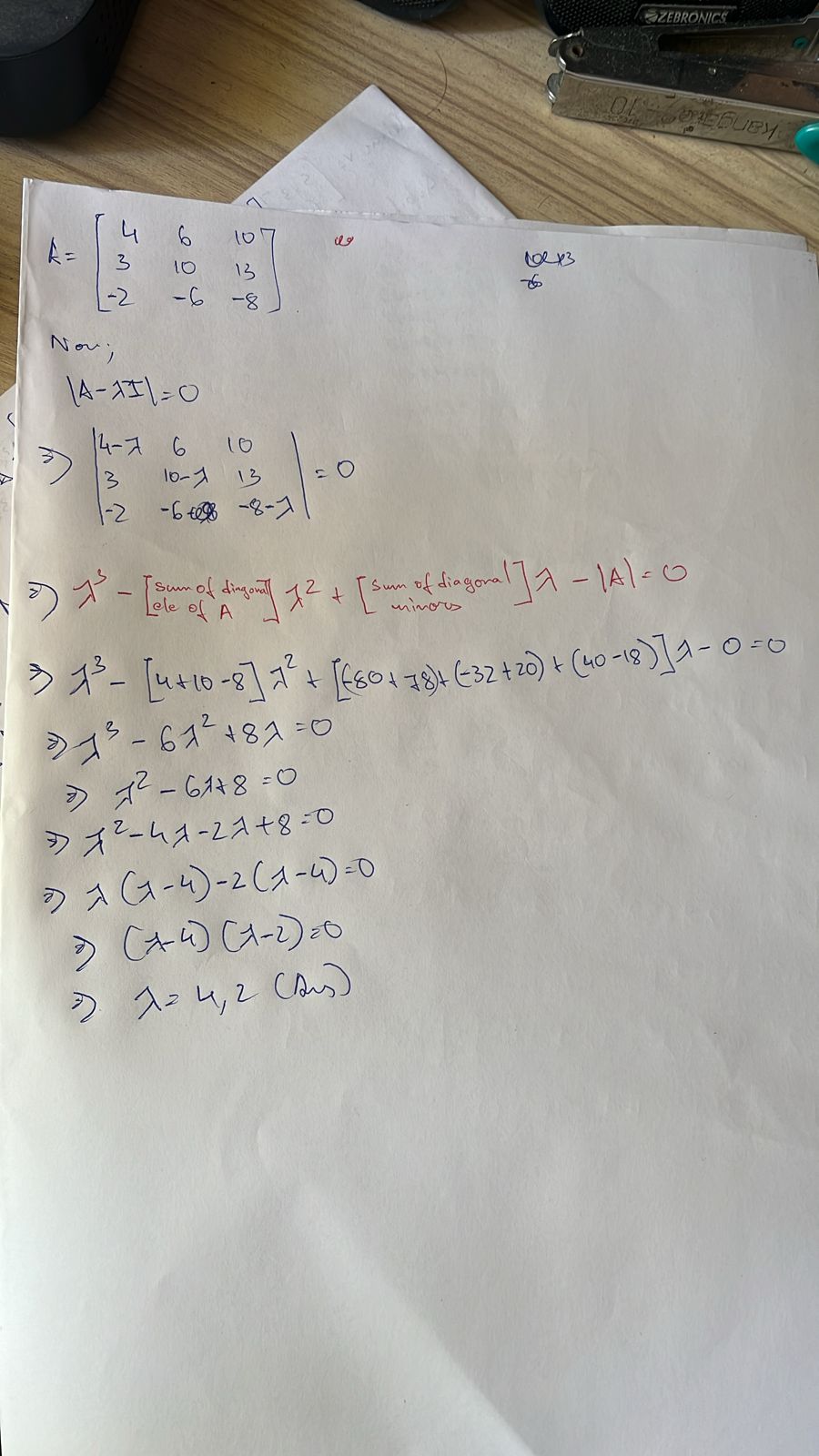


4)Transform the following expression: (A + B) (C + D), Using only NOR gate.

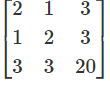


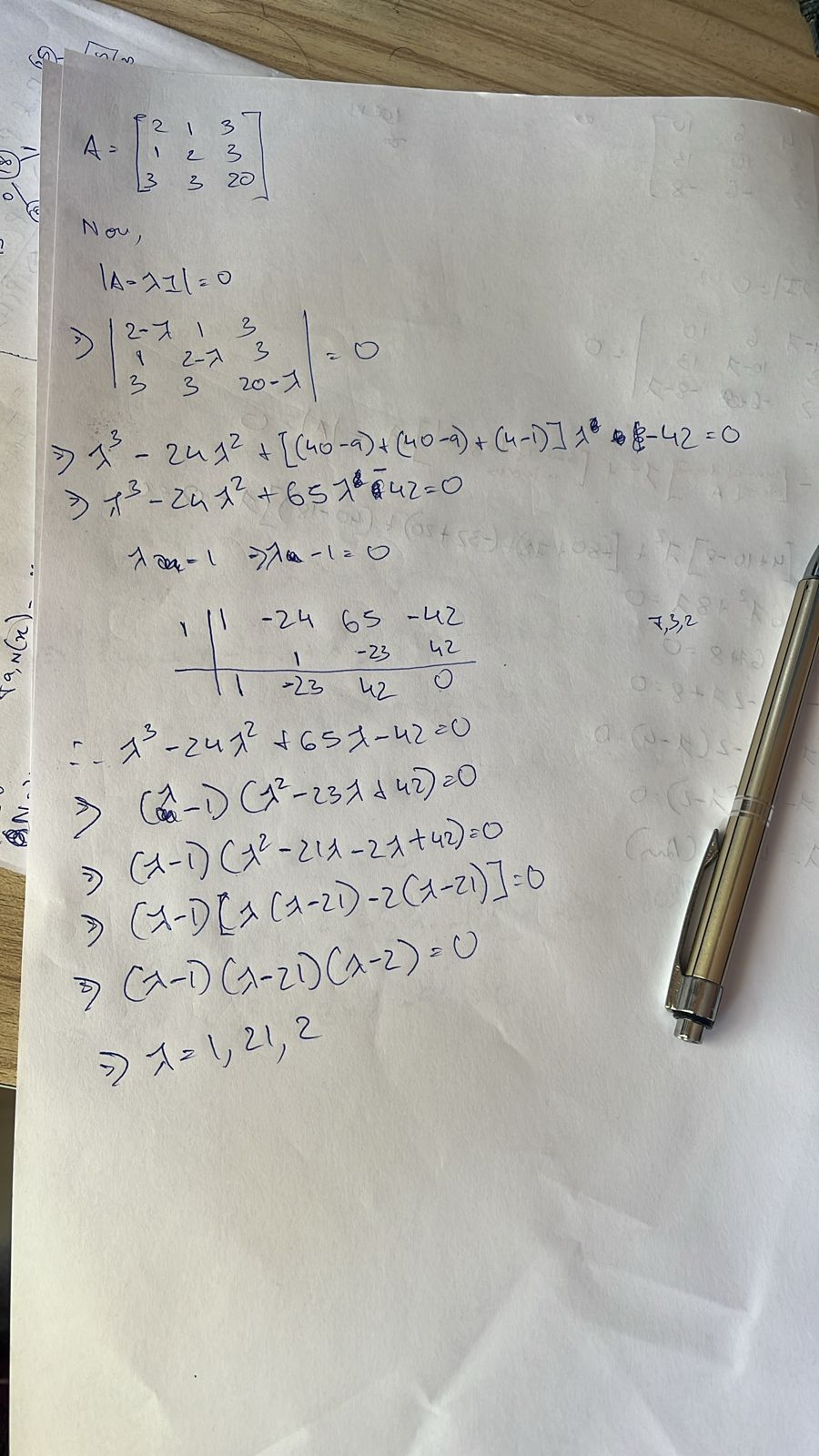
4)Find eigen values for the given matrix.





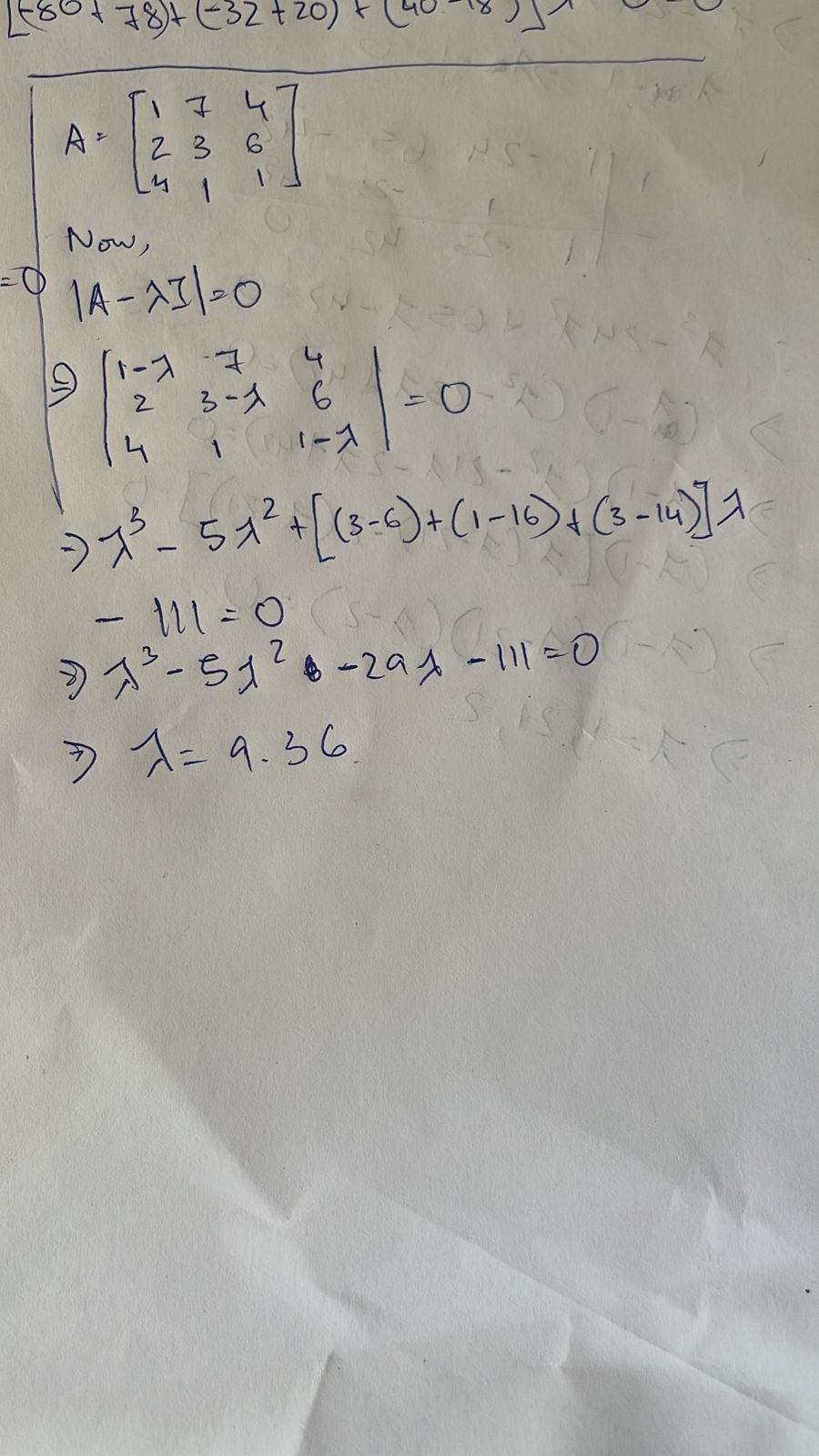
5)Find eigen values for the given matrix.





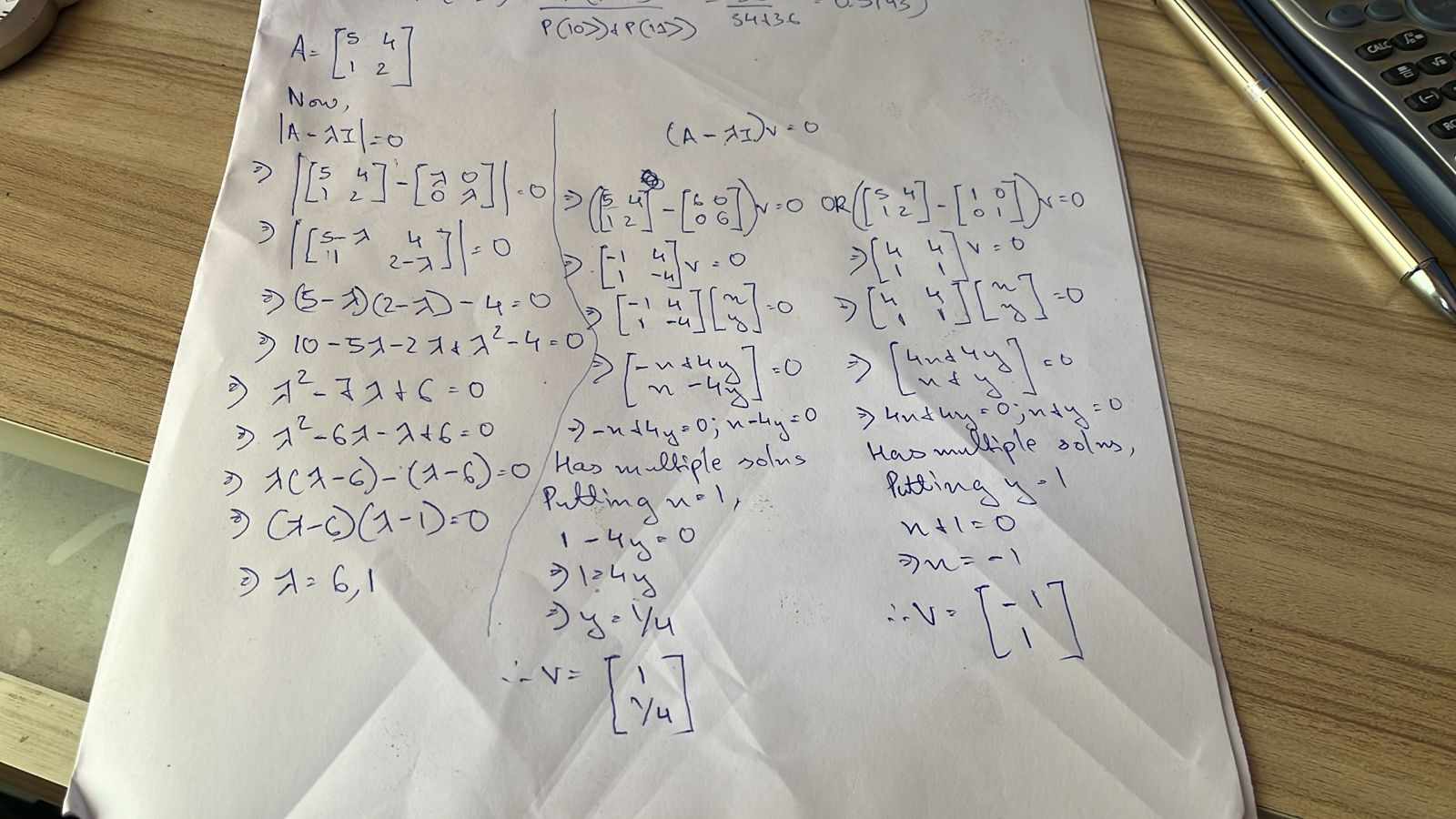
6)Find eigen values for the given matrix.



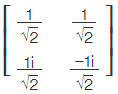


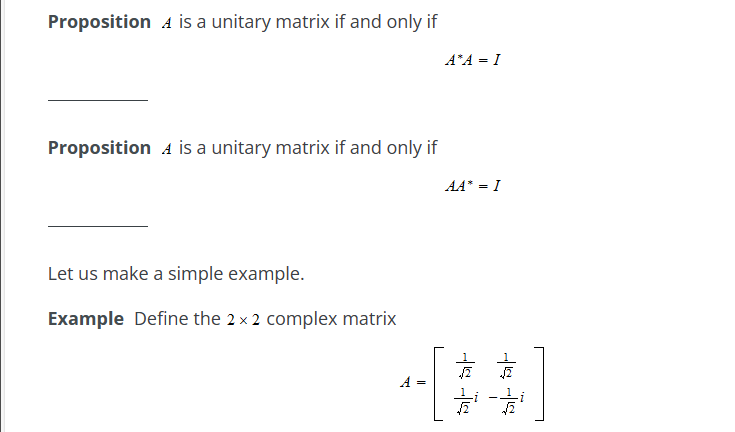
7)Find eigen values and eigen vector for the given matrix.

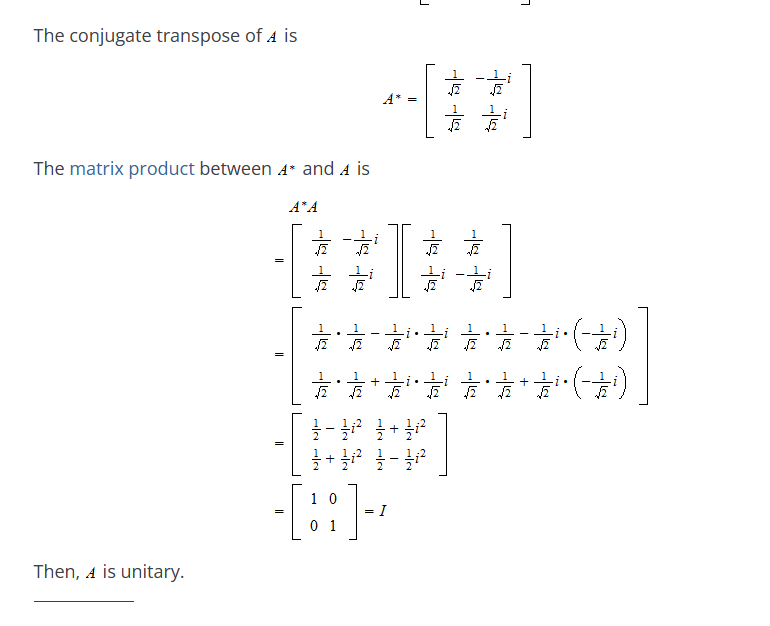




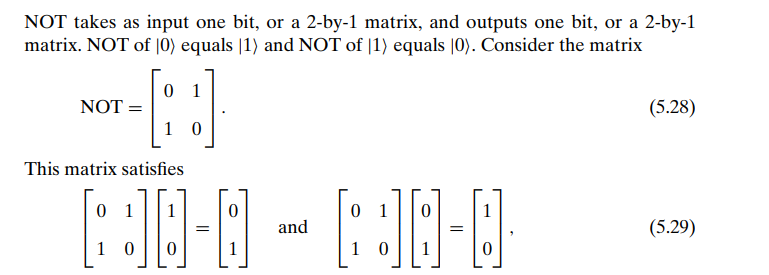
8)Show that given matrix is unitary.

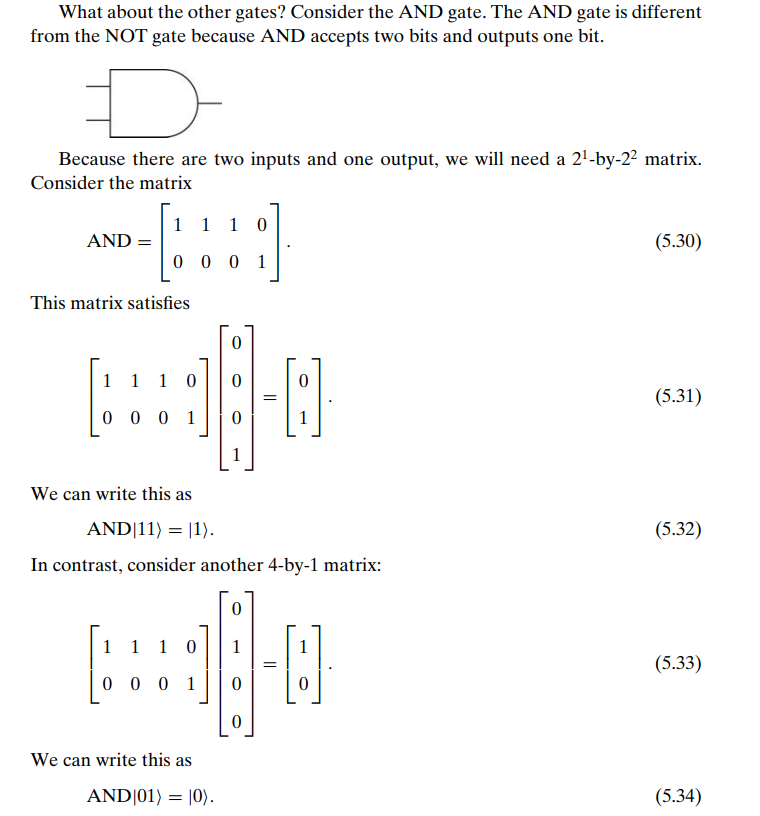


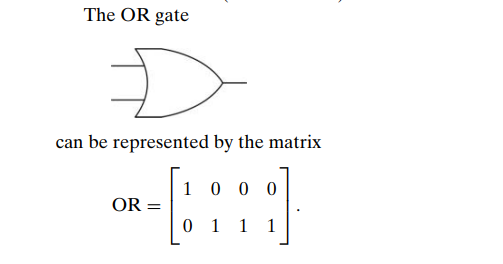


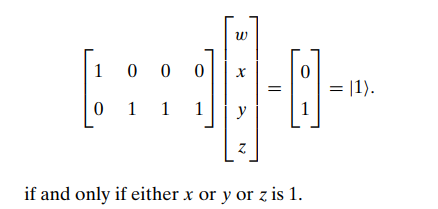


9)Express input and output of basic gates as matrix.

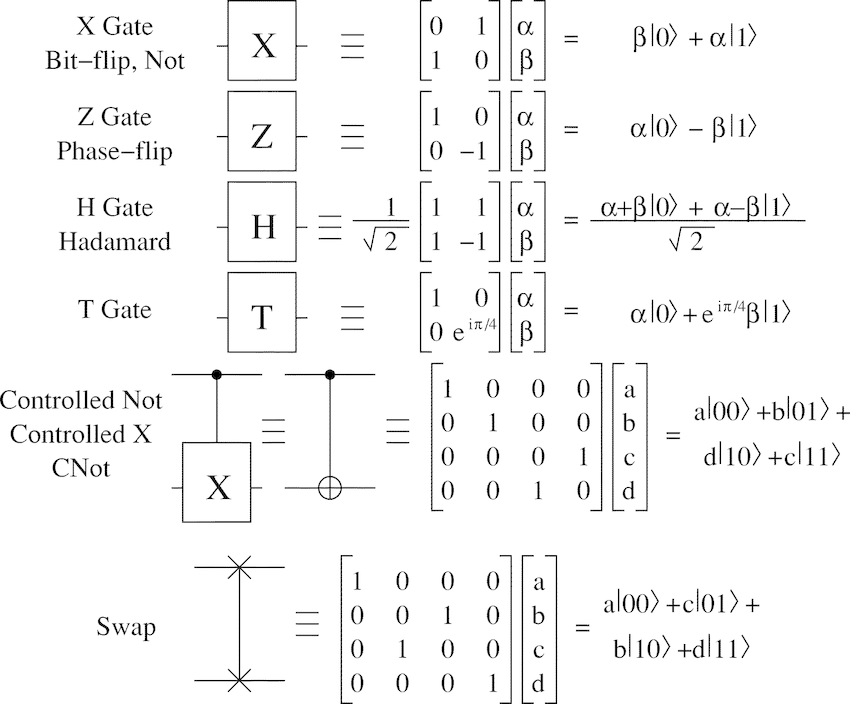






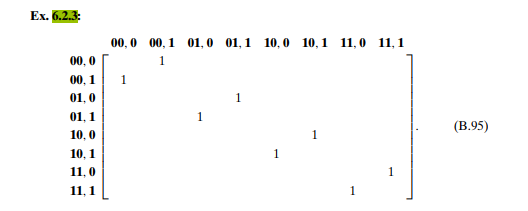


10)Discuss different quantum gates.



11)Consider the function from {0, 1}^2 to {0, 1} that always outputs a 1.

Give the corresponding 8-by-8 unitary matrix.



12)Explain Deutsch algorithm.

The Deutsch algorithm is a fundamental concept in quantum computing that demonstrates its potential superiority over classical algorithms for specific tasks. Here's a breakdown of the algorithm:

**Problem:**

The Deutsch algorithm aims to determine whether a single-bit function f: {0, 1} -> {0, 1} is constant (always outputs the same value) or balanced (outputs 0 for some inputs and 1 for others).

**Classical Approach:**

Requires at least two function evaluations:

Evaluate f(0).

If f(0) is 0, evaluate f(1) to check for a change in output (balanced).

If f(0) is 1, any other output (0 or 1) confirms a balanced function.

In the worst case (constant function), two evaluations are necessary.

**Quantum Advantage:**

The Deutsch algorithm can solve this problem with only one function evaluation using the principles of superposition and interference in quantum mechanics.

**Steps:**

**Qubit Initialization:** Prepare a single qubit in the superposition state: |psi> = 1/sqrt(2) |0> + 1/sqrt(2) |1>

**Oracle Function:** Apply a quantum oracle (black box) that implements the unknown function f. This operation transforms the state based on the function's output for the input state.

If f is constant (always 0 or always 1), both |0> and |1> basis states are mapped to the same output state.

If f is balanced, the |0> and |1> basis states are mapped to different output states.

**Hadamard Transform:** Apply a Hadamard gate (H) to the qubit. This transformation rotates the basis states, creating a superposition of all possible measurement outcomes (0 and 1).

**Measurement:** Measure the qubit in the computational basis (|0> or |1>).

**Outcome Analysis:**

**Constant Function:**

Regardless of the initial state, the oracle maps it to the same state after applying f.

The Hadamard transform then creates an equal superposition of 0 and 1.

Measuring the qubit will result in either 0 or 1 with a 50% probability for each.

**Balanced Function:**

The oracle maps the initial superposition state to a different state depending on the basis state (|0> or |1>).

The Hadamard transform interferes with these states, potentially canceling out one of the basis states in the superposition.

Measuring the qubit will result in a definite outcome (either 0 or 1) with a 100% probability.

**Conclusion:**

By observing the measurement outcome:

**Deterministic outcome (0 or 1):** The function is balanced.

**Probabilistic outcome (50% chance of 0 or 1):** The function is constant.

**Significance:**

The Deutsch algorithm showcases the power of quantum computation. With just one function evaluation, it solves a problem that requires at least two evaluations classically. This highlights the potential of quantum algorithms for specific tasks.

13)Explain Deutsch-Jozsa algorithm.

The Deutsch-Jozsa algorithm is an extension of the Deutsch algorithm and another example of how quantum computers can outperform classical algorithms for specific problems. Here's a breakdown of the Deutsch-Jozsa algorithm:

**Problem:**

The Deutsch-Jozsa algorithm aims to determine the type of a black-box function f: {0, 1}^n -> {0, 1}, where n is an arbitrary number of input bits. The function can be one of two types:

**Constant:** Always outputs the same value (either 0 or 1) regardless of the input.

**Balanced:** Outputs 0 for half of the possible inputs and 1 for the other half.

**Classical Approach:**

Requires n+1 function evaluations in the worst case:

Evaluate f for different input combinations until you observe both 0 and 1 as outputs. This can take up to n+1 evaluations for a balanced function.

**Quantum Advantage:**

The Deutsch-Jozsa algorithm can solve this problem with only one function evaluation using quantum principles.

**Steps:**

**Qubit Initialization:** Prepare n qubits in the superposition state: |psi> = 1/sqrt(2^n) (|00...0> + |00...1> + ... + |11...1>). This creates a superposition of all possible input combinations for the n-bit function.

**Oracle Function:** Apply a quantum oracle that implements the unknown function f. This operation transforms the state based on the function's output for each possible input state in the superposition.

**Hadamard Transform:** Apply a Hadamard gate (H) to each of the n qubits. This transformation creates a superposition of all possible measurement outcomes (0 and 1) for each individual qubit.

**Measurement:** Measure each qubit in the computational basis (|0> or |1>).

**Outcome Analysis:**

**Constant Function:**

Regardless of the initial state, the oracle maps all basis states in the superposition to the same state after applying f.

The Hadamard transform creates an equal superposition of 0s and 1s in each qubit.

Measuring each qubit will result in either 0 or 1 with a 50% probability for each.

**Balanced Function:**

The oracle maps different basis states in the superposition to different output states based on the input bits.

The Hadamard transform interferes with these states, causing one output state (either 0 or 1) to cancel out for each qubit in the superposition.

Measuring any qubit will result in a definite outcome (either 0 or 1) with a 100% probability.

**Conclusion:**

By observing the measurement outcome:

**All qubits collapse to the same state (0 or 1):** The function is constant.

**Any single qubit collapses to a definite state (0 or 1):** The function is balanced.

**Significance:**

The Deutsch-Jozsa algorithm demonstrates the power of quantum computation for solving certain problems exponentially faster than classical algorithms. This highlights the potential of quantum algorithms for tasks involving black-box functions or hidden patterns in data.

**Additional Notes:**

* The Deutsch-Jozsa algorithm becomes particularly significant for larger values of n (number of input bits). As n increases, the classical approach requires exponentially more evaluations, while the quantum approach remains constant with only one evaluation.
* The algorithm relies on perfect implementation of the oracle and error-free quantum operations, which are challenges in current quantum technology.

14)Explain Simon's Periodicity Algorithm.

The Simon's Algorithm is another quantum algorithm that showcases the power of quantum computation for specific tasks. Here's a breakdown of the algorithm:

**Problem:**

The Simon's problem deals with a function f: {0, 1}^n -> {0, 1}^n, where n is the number of input bits. The function is promised to be periodic, meaning there exists a secret string "s" (of length k, k <= n) such that:

* f(x) = f(x ⊕ s) for all input values x. (⊕ denotes bitwise XOR)

The goal is to find a non-zero period "s" (secret string) of the function f.

**Classical Approach:**

* Finding the period classically can be challenging, especially for larger n and k. It might involve trying different input combinations and analyzing the outputs to find a pattern.

**Quantum Advantage:**

The Simon's Algorithm tackles this problem with a significant speedup compared to classical approaches, leveraging quantum superposition and interference.

**Steps:**

* **Qubit Initialization:** Prepare two qubits in the superposition state: |psi> = 1/sqrt(2^n) (|00...0> + |00...1> + ... + |11...1>). This creates a superposition of all possible input combinations for the n-bit function.
* **Oracle Function:** Apply a quantum oracle that implements the unknown function f. This operation transforms the two qubits based on the function's output for the combined input state.
* **Second Qubit Hadamard Transform:** Apply a Hadamard gate (H) to the second qubit only. This creates a superposition of all possible measurement outcomes (0 and 1) for the second qubit.
* **Measurement:** Measure both qubits in the computational basis (|0> or |1>).

**Outcome Analysis:**

The measurement outcome provides information about the secret period "s":

* The first qubit:
  + Doesn't reveal any direct information about "s" but helps identify a period by influencing the second qubit's outcome.
* The second qubit:
  + Due to the Hadamard transform and the periodic nature of f, the measurement outcome will be a string that satisfies the equation f(x) = f(x ⊕ s) for some input value x.
  + This outcome provides a clue about the period "s".

**Period Finding:**

* Repeat steps 1-4 multiple times with fresh qubits.
* Each measurement outcome from the second qubit provides an equation related to the period "s".
* By analyzing the collected equations (system of equations), you can solve for the secret string "s" that satisfies all the equations. This effectively reveals the period of the function.

**Significance:**

The Simon's Algorithm demonstrates the power of quantum computation for solving specific problems involving hidden periodicities. It offers an exponential speedup compared to classical approaches, especially for larger k (length of the secret string). This has potential applications in areas like cryptography and code-breaking.

**Additional Notes:**

* The success of the algorithm relies on obtaining enough independent measurement outcomes to solve for the period efficiently.
* The practical implementation of the algorithm requires careful control of quantum operations and mitigation of errors.

15)Explain Shor's factoring algorithm.

Shor's algorithm is a groundbreaking quantum algorithm that efficiently factors large integers. Factoring large integers is a computationally expensive task for classical computers, but Shor's algorithm offers a significant advantage using the principles of quantum mechanics.

**Problem:**

The goal is to find the prime factorization of a large positive integer N. Prime factors are the building blocks of any composite number (a number divisible by more than 1 and itself).

**Classical Difficulty:**

* Classical algorithms for factoring, like trial division or the quadratic sieve, become increasingly time-consuming as the number of bits in N (its size) increases.
* For large cryptographic keys used in modern encryption, these classical methods become impractical.

**Quantum Advantage:**

Shor's algorithm leverages quantum superposition and the concept of period finding to efficiently factor large integers.

**Steps:**

* **Choose a Random Integer (a):** Select a random integer 'a' less than N. This value is used throughout the algorithm.
* **Quantum Circuit:** The core of the algorithm involves a quantum circuit with the following elements:
  + **Qubit Initialization:** Prepare two qubits in superposition: |psi> = 1/sqrt(2^n) (|00...0> + |00...1> + ... + |11...1>). This creates a superposition of all possible values from 0 to 2^n-1 (n being the number of bits in N).
  + **Modular Exponentiation Oracle:** This oracle implements the function f(x) = a^x mod N. It computes the modular exponentiation of 'a' raised to the power of each possible value in the superposition (represented by the first qubit) modulo N.
  + **Quantum Fourier Transform (QFT):** This operation applies the Quantum Fourier Transform to the first qubit. It transforms the superposition of states into a superposition of frequencies related to the period of the function f(x).
* **Measurement:** Measure both qubits in the computational basis (|0> or |1>).

**Period Finding:**

* The measured value in the second qubit doesn't directly reveal the factors of N.
* However, the outcome from the first qubit (after QFT) provides information about the period of the function f(x). In Shor's algorithm, this period is related to the factors of N.
* By repeating steps 1-3 multiple times (with fresh qubits each time) and analyzing the obtained periods, you can use mathematical techniques to compute the prime factors of N.

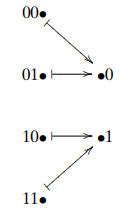
**Significance:**

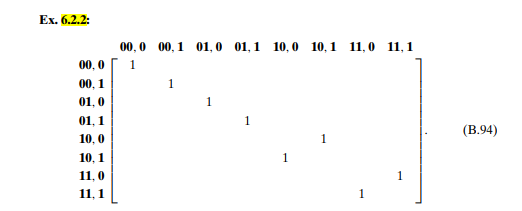
Shor's algorithm offers an exponential speedup compared to classical factoring algorithms for large integers. This has significant implications for cryptography, where many encryption schemes rely on the difficulty of factoring large numbers. If Shor's algorithm can be implemented efficiently on large-scale quantum computers, it could potentially break these encryption methods.

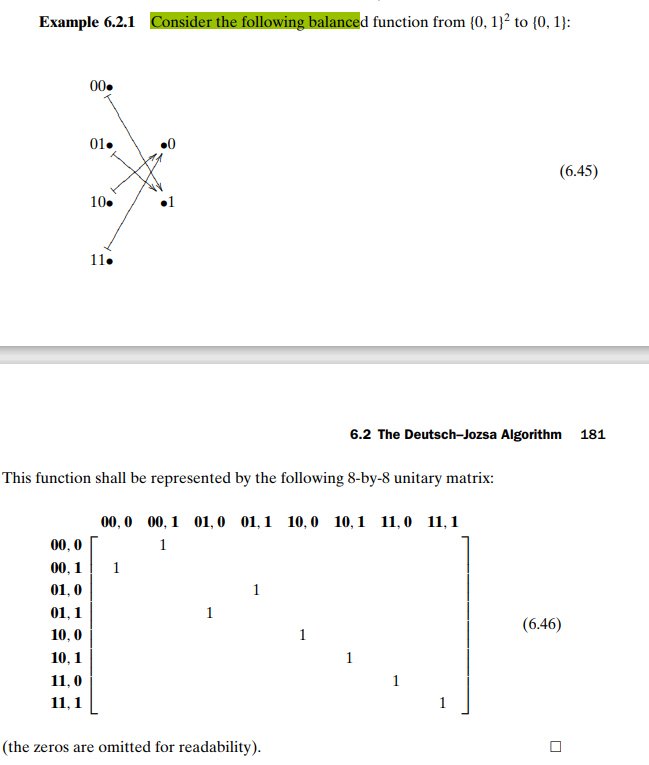
**Additional Notes:**

* Shor's algorithm is a complex algorithm with several intricacies.
* Building large-scale quantum computers capable of running Shor's algorithm efficiently is an ongoing research area.
* While Shor's algorithm poses a threat to certain encryption methods, it also opens doors for developing new, potentially more secure quantum-resistant cryptography.

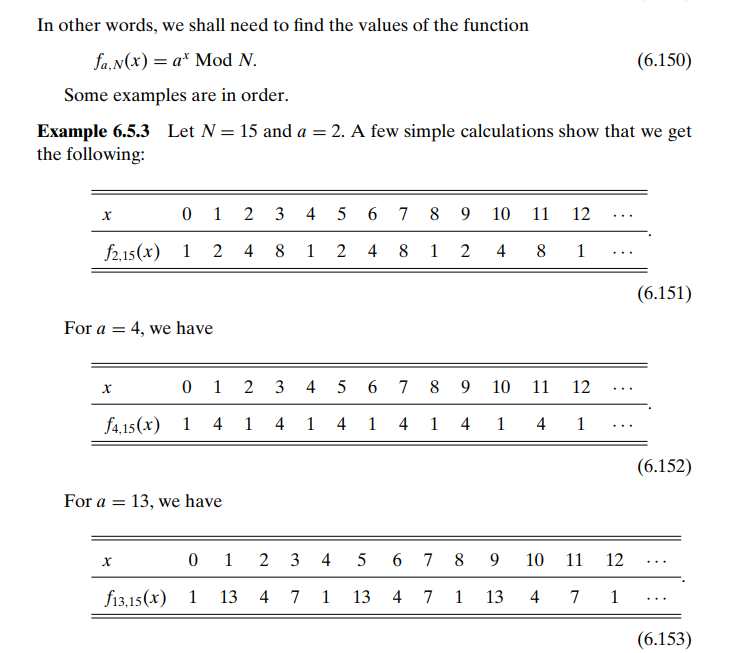
16)Consider the following balanced function from {0, 1}^2 to {0, 1} and represent it as a matrix.



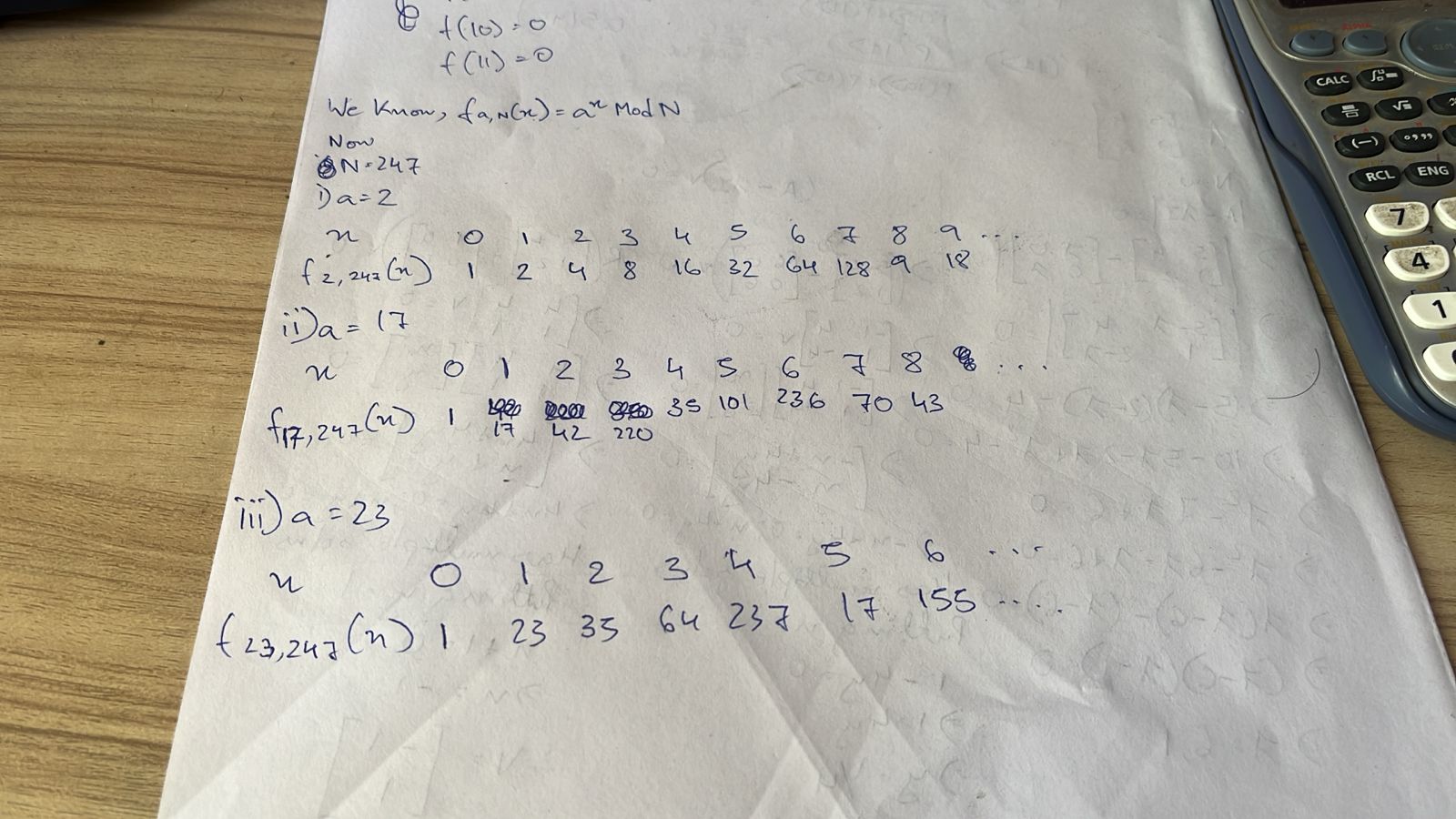




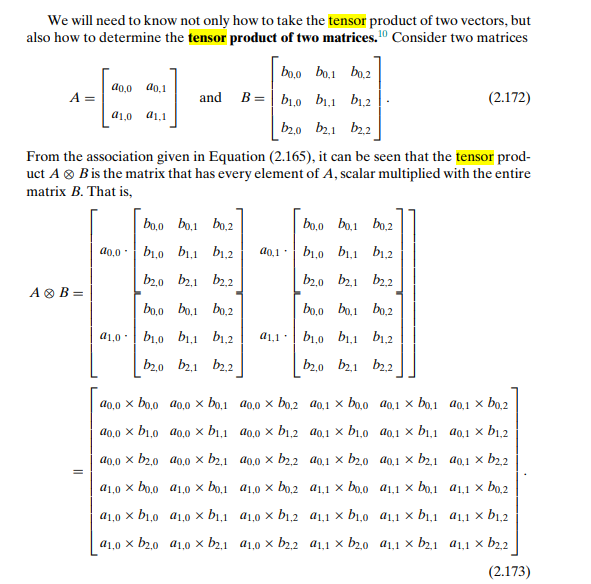
17)Let N = 15 and a = 2, 4, 13. Apply Shor’s algorithm to find f(a,N).

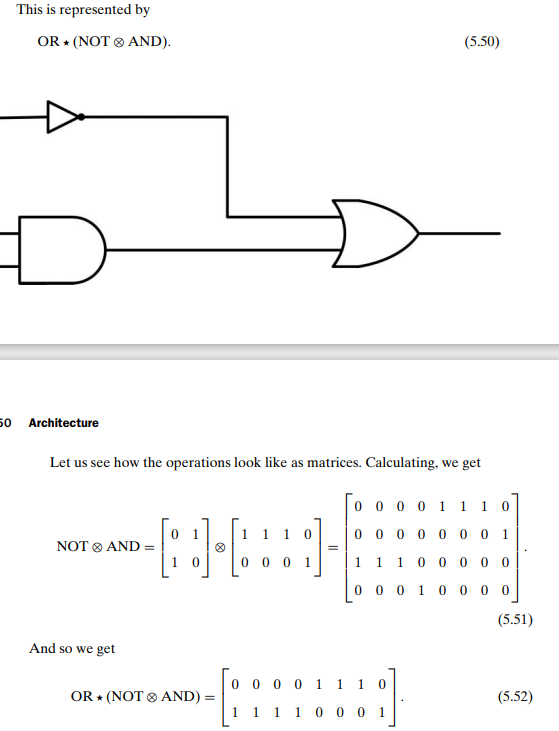


18)Calculate the first few values of f(a,N) for N = 247 and (i) a = 2 (ii) a = 17(iii) a = 23.

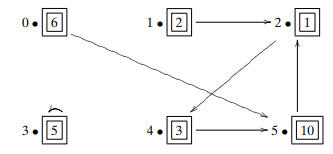


19)Consider a circuit that is represented by OR \*(NOT⊕AND). Find its corresponding matrix operation.

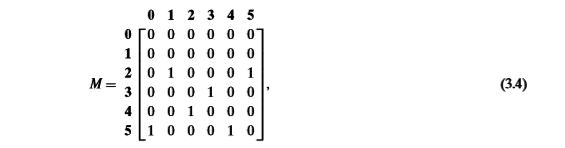




20)Consider the given graph for a dynamic system and describe it as matrix.

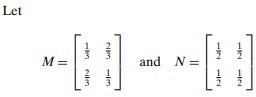


This graph is easily stored in memory as a Boolean adjacency matrix:

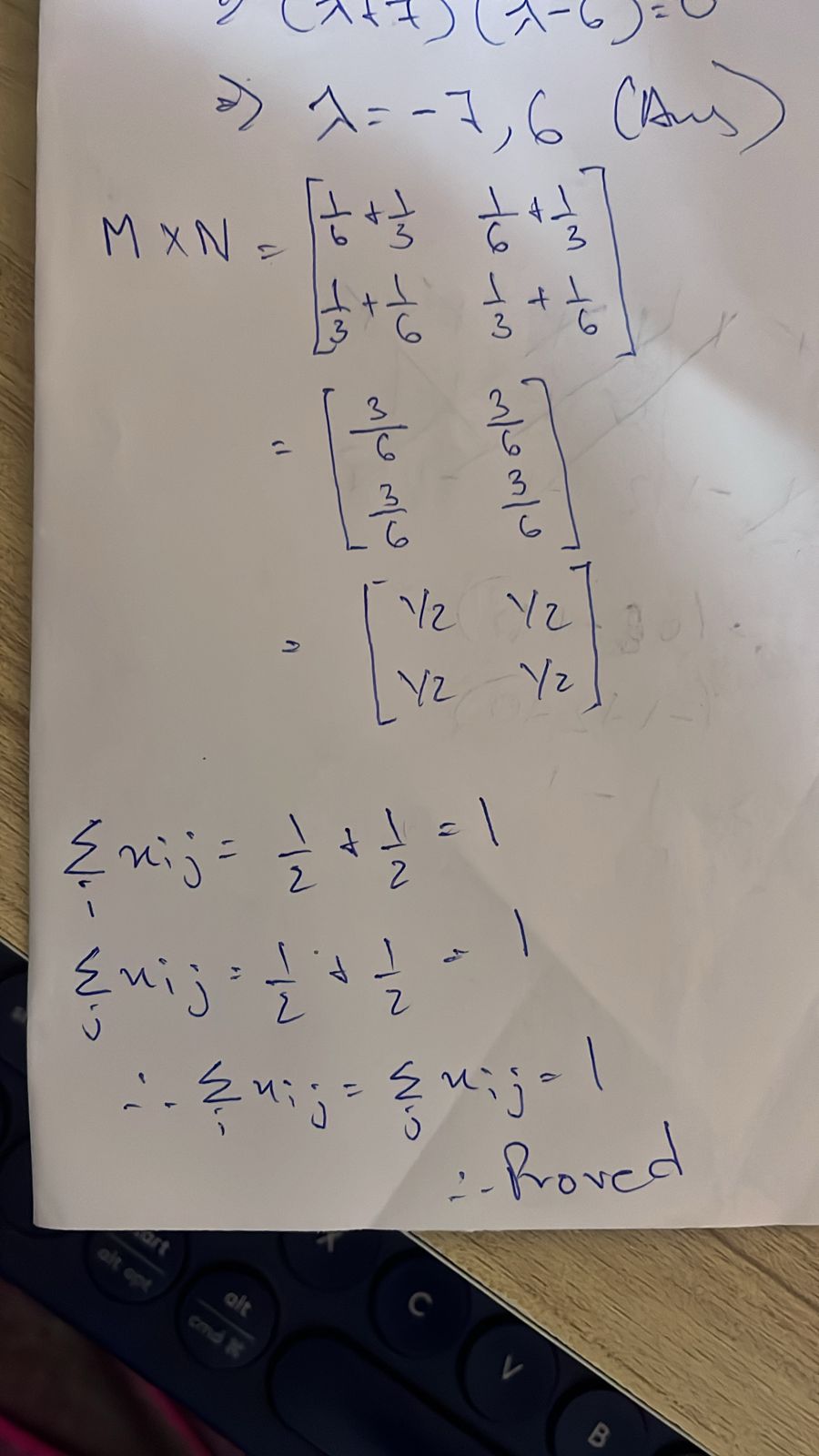


where M [i, j] = 1 if and only if there is an arrow from vertex i to vertex j. The requirement that each vertex has exactly one outgoing edge corresponds to the fact that each column of the Boolean adjacency matrix contains exactly one 1.

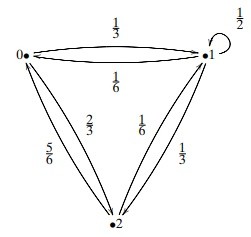
21)Multiply M\*N and show the result is doubly stochastic matrix.



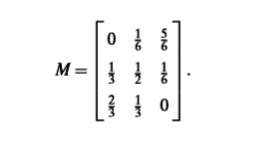


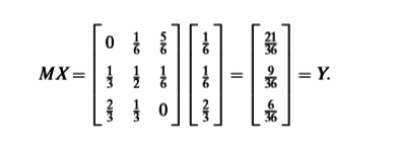


22)Consider the given graph and a specific state : probability is 1/ 6 that the marble is on vertex 0, the probability is 1/ 6 that the marble is on vertex 1, and the probability is 2/3 that the marble is on vertex 2. Calculate how a state changes.



The adjacency matrix of this graph is:





23)Explain the quantum billiard ball with proper graph and matrix.

Explain the quantum billiard ball with proper graph and matrix.

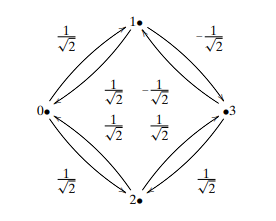
**Conceptualizing the Billiard Ball:**

* Imagine a billiard ball confined to a rectangular box.
* In classical mechanics, the ball has a definite position and momentum (velocity).
* In quantum mechanics, the ball's location and momentum become probabilities described by a wave function.

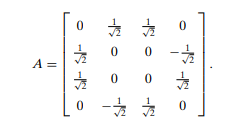
**Graph Representation:**

A common approach to represent the quantum billiard ball uses a discrete-space model. The box is divided into a grid of squares, and the possible states of the ball correspond to its location on the grid.

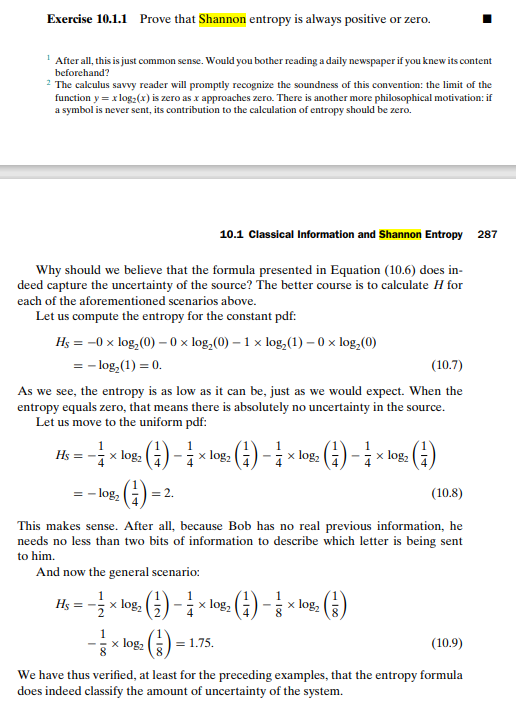
* Nodes: Each square in the grid represents a node in the graph.
* Edges: Edges connect neighboring squares, indicating the possibility of the ball moving between them.

**Matrix Representation:**

The system's behavior can be represented by a Hamiltonian matrix (H). This matrix describes the energy transitions between different states (nodes) of the ball.

* Diagonal elements (H\_ii): Represent the potential energy of the ball at a particular node (square) due to confinement within the box. Typically, these are set to a constant value.
* Off-diagonal elements (H\_ij): Represent the kinetic energy associated with the ball moving between neighboring nodes (squares). These are typically set to a negative value (-t) to indicate the energy difference between adjacent squares.

24)Prove that Shannon entropy is always positive or zero.



25)Justify von Neumann entropy.

**What is von Neumann Entropy?**

* **Definition:** The von Neumann entropy, S, is a generalization of Shannon entropy to the realm of quantum mechanics. It measures the amount of uncertainty or randomness associated with a quantum state represented by a density matrix ρ:

S(ρ) = −Tr(ρ log₂(ρ))

where Tr() denotes the trace of a matrix.

* **Density Matrix:** A density matrix is a mathematical tool to describe both pure quantum states (those that have definite properties) and mixed states (statistical mixtures of pure states).

**Justifications**

* **Analogy to Shannon Entropy:**
  + Von Neumann entropy reduces to Shannon entropy for classical probability distributions, where the density matrix becomes diagonal.
  + It shares many of the same properties as Shannon entropy, including non-negativity, reflecting the uncertainty in a system.
* **Thermodynamic Connection:**
  + Von Neumann entropy has a strong link to thermodynamic entropy. Under certain conditions, it can be shown to be proportional to thermodynamic entropy, providing a bridge between quantum information theory and thermodynamics.
* **Entanglement Measure:**
  + Von Neumann entropy of subsystems of a larger quantum system is a fundamental measure of quantum entanglement. A higher entropy for a subsystem generally indicates a greater degree of entanglement with the rest of the system.
* **Unitary Invariance:**
  + Von Neumann entropy is invariant under unitary transformations, which are the fundamental operations in quantum mechanics. This means it doesn't change under valid quantum evolutions, making it a robust measure of intrinsic uncertainty.
* **Operational Interpretation:**
  + Von Neumann entropy can be related to the number of qubits needed to faithfully represent a mixed quantum state. This gives it meaning in terms of quantum computation and information processing.

26)Explain data compression techniques.

Data compression involves reducing the size of data files without significantly affecting their usability. The techniques can be broadly classified into two categories:

**Lossless Compression:**

* **Run-Length Encoding (RLE)**: This method replaces sequences of the same data values within a file with a count number and a single value.
* **Huffman Coding**: It uses variable-length codes for different characters based on their frequencies. More frequent characters have shorter codes.
* **Lempel-Ziv-Welch (LZW)**: It builds a dictionary of byte patterns as it reads through the data, replacing recurring patterns with shorter codes.

**Lossy Compression:**

* **Transform Coding**: This is often used in image compression (like JPEG) where less important information is discarded.
* **Quantization**: In audio and image compression, it reduces the precision of certain components to save space.
* **Subsampling**: This technique reduces the resolution of images by discarding some of the color information, often imperceptible to humans.

27)Explain Huffman coding and its application in data compression process.

Huffman coding is a popular method for lossless data compression that’s based on the frequency of occurrence of a data item (like characters in a file). The process involves creating a Huffman tree where the most frequent items are placed near the root, and the least frequent are placed farthest from the root. Here’s how it works:

* **Frequency Table**: First, count the frequency of each data item in the input.
* **Huffman Tree**: Build a binary tree where each leaf node represents a data item. The path from the root to a leaf gives the Huffman code for that item.
* **Encoding Data**: Use the Huffman tree to encode data. More frequent items will have shorter codes.
* **Decoding Data**: To decode, follow the paths in the Huffman tree based on the bits in the encoded data until you reach a leaf node.

**Application in Data Compression:** Huffman coding is widely used in file compression formats like ZIP and GZIP, as well as multimedia codecs like JPEG and MP3. The method is particularly effective when there are clear differences in frequency, with some items occurring much more often than others. By using shorter codes for common items, Huffman coding reduces the overall size of the data, leading to efficient storage and transmission. It’s crucial in applications where preserving the original data is important, as the process is fully reversible without any loss of information.

28) Explain error-correcting codes.

Error-correcting codes (ECC) are methods in coding theory used to detect and correct errors in data transmission or storage. They are crucial for ensuring the reliability of digital communication and storage systems, especially in environments prone to interference or corruption. Here’s a brief overview:

**How ECC Works:**

* **Encoding**: Before data is sent or stored, it is encoded using an algorithm that adds redundant data, known as parity bits.
* **Transmission/Storage**: The encoded data, now larger than the original, is transmitted or stored.
* **Detection**: The system checks the received or retrieved data using the redundancy to detect any discrepancies.
* **Correction**: If errors are detected, the system uses the redundant bits to determine the location and nature of the errors and correct them.

**Types of ECC:**

* **Hamming Codes**: Can detect and correct single-bit errors, and detect double-bit errors.
* **Reed-Solomon Codes**: Widely used in digital communications and storage, capable of correcting burst errors.
* **Convolutional Codes**: Often used in wireless communication, they use the data’s history to correct errors.
* **Turbo Codes and LDPC (Low-Density Parity-Check) Codes**: Provide near-Shannon limit error correction performance.

**Applications of ECC:**

* **Digital Storage**: ECC is used in RAM, hard drives, and solid-state drives to correct transient errors.
* **Wireless Communication**: Mobile phones, satellite communication, and Wi-Fi all use ECC to maintain data integrity.
* **Broadcasting**: Digital TV and radio use ECC to ensure clear reception despite signal degradation.

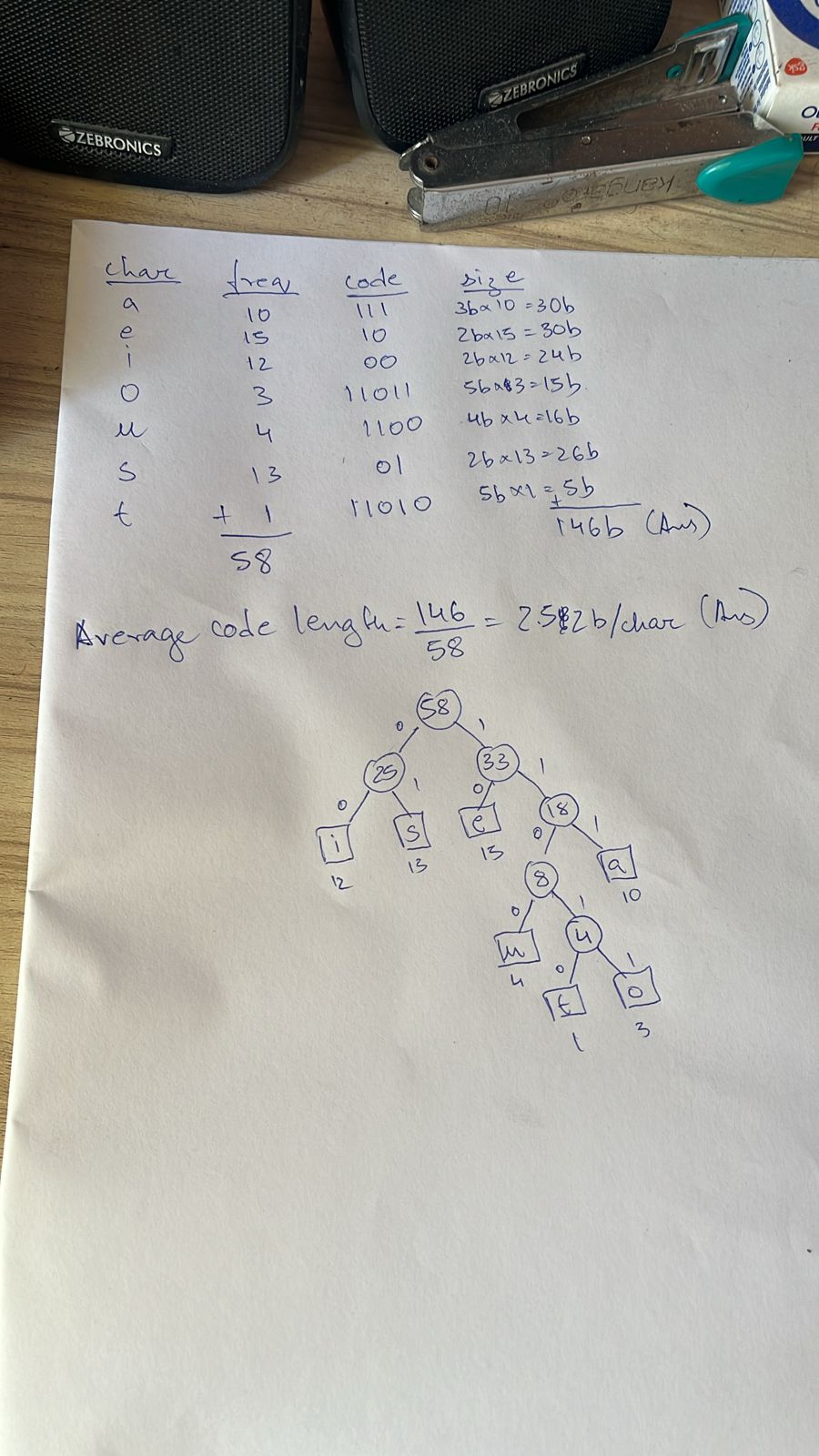
ECC is a balance between redundancy and efficiency; adding more redundancy allows for better error correction but requires more bandwidth or storage space. The design of an ECC system depends on the expected error rates and the importance of data integrity for the specific application.

29)A file contains the following characters with the frequencies as shown. If Huffman Coding is used for data compression, determine-

a.Huffman Code for each character Average code length

b. Length of Huffman encoded message (in bits)

|  |  |
| --- | --- |
| Characters | Frequencies |
| a | 10 |
| e | 15 |
| i | 12 |
| o | 3 |
| u | 4 |
| s | 13 |
| t | 1 |



30)Reframe error-correcting codes.

Error-correcting codes are a significant concept in quantum computing, as they are essential for maintaining the integrity of information. In the context of quantum computing, error-correcting codes are used to protect quantum information against decoherence and other quantum noise. They are a crucial part of quantum information theory, which is a field that extends the classical information theory into the quantum domain. These codes are designed to correct errors without measuring the quantum state directly, which would otherwise collapse the state and destroy the information. The development and implementation of robust error-correcting codes are vital for the advancement of reliable quantum computers.