1. Print your name on the screen
2. Read your name, age, email and print them on the screen
3. Read an integer and print it on the screen.
4. Read an float and print it on the screen.
5. Read a double and print it on the screen.
6. Read a character and print it on the screen.
7. Read a long number and print it on the screen
8. Read a bool value and print it on the screen
9. Write a program to sum two numbers and print the sum on screen.
10. Write a program to subtract two numbers and print the result.
11. Write a program to multiply two numbers and print the result.
12. Write a program to divide two numbers and print the result.
13. Write a program to calculate the remainder of two numbers.
14. Write a program to calculate the half of a number
15. Read two strings from the user and print them
16. Read three character from the user and print them
17. Read three numbers from the user and print them
18. Read your name and age , then print the length of your name.
19. Read your name and age then determine your age category based on your age
20. Write a program to read your username and password and print them on the screen.
21. Write a program to read your username and password for login purpose with predefined data.
22. Write a program to calculate the square root of a number.
23. Write a program to calculate the logarithm of a number.
24. Write a program to calculate the sine of an angle.
25. Write a program to calculate the cosine of an angle.
26. Write a program to calculate the tangent of an angle.
27. Write a program to convert Celsius to Fahrenheit.
28. Write a program to convert Fahrenheit to Celsius.
29. Write a program to convert kilometers to miles.
30. Write a program to convert miles to kilometers.
31. Write a program to calculate the area of a rectangle.
32. Write a program to calculate the area of a circle.
33. Write a program to calculate the perimeter of a rectangle.
34. Write a program to calculate the perimeter of a circle.
35. Write a program to calculate the area of a triangle.
36. Write a program to calculate the perimeter of a triangle.
37. Write a program to calculate the volume of a sphere.
38. Write a program to calculate the volume of a cube.
39. Write a program to calculate the volume of a cylinder.
40. Write a program to calculate the volume of a cone.
41. Write a program to calculate the area of a trapezoid.
42. Write a program to calculate the area of a parallelogram.
43. Write a program to calculate the area of a rhombus.
44. Write a program to calculate the area of a pentagon.
45. Write a program to calculate the area of a hexagon.
46. Write a program to calculate the area of a regular polygon.
47. Write a program to calculate the perimeter of a regular polygon.
48. Write a program to check if a number is in a ragne.
49. Write a program to check if a number is positive, negative, or zero.
50. Write a program to check if a number is even or odd.
51. Write a program to check if a year is a leap year or not.
52. Write a program to check if a character is a vowel or consonant.
53. Write a program to check if a number is prime or not.
54. Write a program to find the largest among three numbers.
55. Write a program to find the smallest among three numbers.
56. Write a program to display the name of the day of the week based on its corresponding number.
57. Write a program to display the name of a month based on its corresponding number.
58. Write a program to calculate the number of days in a given month.
59. Write a program to display the name of a shape based on its number of sides.
60. Write a program to calculate the area of a shape based on its type.
61. Write a program to display the name of a color based on its code.
62. Write a program to convert a number into its corresponding Roman numeral.
63. Write a program to calculate the grade based on a student's score.
64. Write a program to display the name of a country based on its ISO code.
65. Write a program to perform basic calculator operations (+, -, \*, /) based on user input.
66. Write a program to convert a given temperature from Celsius to Fahrenheit or vice versa based on user input.
67. Write a program to display the name of a day in a week based on its abbreviation.
68. Write a program to display the name of a programming language based on its file extension.
69. Write a program to calculate the discount amount based on a customer's membership level.
70. Write a program to display the name of a planet based on its position from the sun.
71. Write a program to perform different actions based on the user's choice (e.g., play, pause, stop) in a music player.
72. Write a program to calculate the net salary of an employee based on their designation.
73. Write a program to display the name of a continent based on a given country.
74. Write a program to calculate the age group based on a person's age.
75. Write a program to calculate the tax amount based on a person's income bracket.
76. Write a program to display the name of a fruit based on its color.
77. Write a program to calculate the commission amount based on a salesperson's performance.
78. Write a program to display the name of a musical instrument based on its category.
79. Write a program to display the name of a vehicle based on its type (e.g., car, bike, truck).
80. Write a program to display the name of a currency based on a given country.
81. Write a program to calculate the shipping cost based on the weight of a package.
82. Write a program to display the name of a drink based on its flavor.
83. Write a program to display the name of a flower based on its color.
84. Write a program to calculate the fare amount based on the distance traveled in a transportation system.
85. Write a program to display the name of a sport based on its type.
86. Certainly! Here are 30 programming problems focused on for loops:
87. 1. Write a program to display the numbers from 1 to 10.
88. 2. Write a program to display the even numbers from 1 to 20.
89. 3. Write a program to calculate the sum of numbers from 1 to 100.
90. 4. Write a program to display the multiplication table of a given number.
91. 5. Write a program to calculate the factorial of a number.
92. 6. Write a program to display the Fibonacci sequence up to a given number of terms.
93. 7. Write a program to find the largest element in an array.
94. 8. Write a program to find the smallest element in an array.
95. 9. Write a program to calculate the average of numbers in an array.
96. 10. Write a program to display the reverse of a given string.
97. 11. Write a program to check if a number is prime.
98. 12. Write a program to find the sum of digits of a number.
99. 13. Write a program to check if a string is a palindrome.
100. 14. Write a program to display the factors of a number.
101. 15. Write a program to calculate the power of a number.
102. 16. Write a program to display the ASCII values of all uppercase letters.
103. 17. Write a program to display the first N natural numbers.
104. 18. Write a program to display the odd numbers from 1 to 50.
105. 19. Write a program to calculate the sum of even numbers from 1 to 100.
106. 20. Write a program to display the squares of numbers from 1 to 10.
107. 21. Write a program to display the cube of numbers from 1 to 10.
108. 22. Write a program to display the multiplication table of numbers from 1 to 5.
109. 23. Write a program to calculate the factorial of numbers from 1 to 10.
110. 24. Write a program to display the sum of digits of numbers from 1 to 50.
111. 25. Write a program to calculate the average of numbers from 1 to 20.
112. 26. Write a program to display the reverse of numbers from 1 to 10.
113. 27. Write a program to check if numbers from 1 to 100 are divisible by 3.
114. 28. Write a program to display the factors of numbers from 1 to 20.
115. 29. Write a program to calculate the power of numbers from 1 to 5.
116. 30. Write a program to display the ASCII values of all lowercase letters.
117. These problems should provide a good practice for using for loops in programming. Happy coding!
118. Write a program to check if a triangle is equilateral, isosceles, or scalene.
119. Write a program to check if a number is a perfect square.
120. Write a program to check if a number is a palindrome.
121. Write a program to check if a number is Armstrong or not.
122. Write a program to check if a string is a palindrome.
123. Write a program to check if a string is an anagram of another string.
124. Write a program to check if a number is a power of two.
125. Write a program to check if a number is a multiple of another number.
126. Write a program to check if a number is divisible by both 3 and 5.
127. Write a program to check if a given year is a century year.
128. Write a program to check if a number is a perfect number.
129. Write a program to check if a number is a strong number.
130. Write a program to check if a number is a happy number.
131. Write a program to check if a number is a narcissistic number.
132. Write a program to check if a number is a Harshad number.
133. Write a program to check if a number is a Smith number.
134. Write a program to check if a number is an abundant number.
135. Write a program to check if a number is a deficient number.
136. Write a program to check if a string is a pangram.
137. Write a program to check if a string is a palindrome ignoring spaces and punctuation.
138. Write a program to check if a year is a leap year using nested if-else statements.
139. Write a program to check if a given time is valid (hours, minutes, and seconds).
140. Write a program to check if a triangle is a right-angled triangle.
141. Write a program to add two numbers.
142. Write a program to subtract two numbers.
143. Write a program to multiply two numbers.
144. Write a program to divide two numbers.
145. Write a program to find the remainder of division between two numbers.
146. Write a program to calculate the square of a number.
147. Write a program to calculate the cube of a number.
148. Write a program to calculate the average of three numbers.
149. Write a program to calculate the factorial of a number.
150. Write a program to calculate the sum of numbers from 1 to n.
151. Write a program to calculate the product of numbers from 1 to n.
152. Write a program to calculate the power of a number.
153. Write a program to find the maximum of two numbers.
154. Write a program to find the minimum of two numbers.
155. Write a program to swap two numbers using a temporary variable.
156. Write a program to swap two numbers without using a temporary variable.
157. Write a program to check if a number is even or odd using bitwise operations.
158. Write a program to check if a number is a multiple of another number.
159. Write a program to check if a number is divisible by both 3 and 5.
160. Write a program to calculate the sum of digits of a number.
161. Write a program to calculate the product of digits of a number.
162. Write a program to calculate the average of an array of numbers.
163. Write a program to find the maximum element in an array.
164. Write a program to find the minimum element in an array.
165. Write a program to sort an array in ascending order.
166. Write a program to sort an array in descending order.
167. Write a program to find the factorial of a number using recursion.
168. Write a program to calculate the Fibonacci sequence.
169. Write a program to find the GCD (Greatest Common Divisor) of two numbers.
170. Write a program to find the LCM (Least Common Multiple) of two numbers.
171. Write a program to check if a number is a prime number.
172. Write a program to generate prime numbers in a given range.
173. Write a program to calculate the sum of all even numbers in a given range.
174. Write a program to calculate the sum of all odd numbers in a given range.
175.  C++ Program to print "Hello, World!".
176.  C++ Program to print an integer entered by the user.
177.  C++ Program to Add/Subtract/Multiply/Divide Two Integers.
178.  C++ Program to Add/Subtract/Multiply/Divide Two Integers entered by the user.
179.  C++ Program to Add/Subtract/Multiply/Divide two Floating Point Numbers.
180.  C++ Program to Compute Quotient and Remainder.
181.  C++ Program to Calculate the Area and Circumference of a Circle.
182.  C++ Program to Calculate the Area of a Scalene Triangle.
183.  C++ Program to Calculate the Area of an Equilateral Triangle.
184.  C++ Program to Calculate the Area of Right Angle Triangle.
185.  C++ Program to Calculate the Area and Perimeter of a Rectangle.
186.  C++ Program to Calculate the Area and Perimeter of a Square.
187.  C++ program that converts between Celsius and Fahrenheit temperatures based on user input. You can also add conversions for Kelvin.
188.  C++ Program to Find ASCII Value of a Character.
189.  C++ Program to Find the Size of int, float, double, and char.
190.  C++ Program to Check Whether a Number is Even or Odd.
191.  C++ Program to Check Whether a Number is Positive or Negative.
192.  C++ Program to Check Whether a Character is a Vowel or Consonant.
193.  C++ Program to find the Largest Number Among Three Numbers.
194.  C++ Program to find if the entered year is a leap year or not.
195.  C++ program that allows the user to perform basic arithmetic operations (addition, subtraction, multiplication, division) on two numbers. You can also add error handling for division by zero.
196.  BMI Calculator: Create a program that calculates a person's Body Mass Index (BMI) based on their weight and height input. Provide a classification of whether the person is underweight, normal weight, overweight, or obese. Use cin, cout. Formula: bmi = weight / (height \* height)

| 1. **bmi < 18.5** | 1. **Underweight** |
| --- | --- |
| 1. bmi < 24.9 | 1. Normal Weight |
| 1. bmi < 29.9 | 1. Overweight |
| 1. Otherwise | 1. Obese |

1.  Nested condition
2. Get the age and membership\_status as user input. membership\_status can be only Y or y. So, if the age is bigger or equal to 18 and if the user is a member of our shop, we provide a 10% discount, else we charge fully.
3. Write a simple chatbot program using nested conditions.
4.  Switch
5. Program to use switch statement. Display Monday to Sunday.
6. Program to display arithmetic operator using switch case.
7.  C++ Program to Swap Two Numbers.
8.  C++ Program to Find all Roots of a Quadratic equation.
9.  C++ Program to Check Whether a Character is an Alphabet or not.
10.  C++ Program to Calculate the Sum of Natural Numbers.
11.  Program to calculate the sum of numbers from m to n.
12. Hint: Input two numbers m and n. Find the sum of all numbers from m to n. For example m=3 and n=8 then sum will be 3 + 4 + 5 + 6 + 7 + 8 = 33.
13.  Program to print Fibonacci series up to 100.
14. Hint: Fibonacci Series is 1, 1, 2, 3, 5, 8, 13, 21, ...
15.  C++ program to print Even numbers up to 100.
16.  C++ program to Generate a Multiplication Table.
17.  C++ program to Calculate the Power of a Number.
18.  **Factorial Calculator:** Write a program that calculates the factorial of a given positive integer. Factorial of a number is the product of all positive integers from 1 to that number.
19.  **Prime Number Checker:** Create a program that determines whether a given number is prime or not. A prime number is a positive integer greater than 1 that has no positive divisors other than 1 and itself.
20.  C++ Program to Display Prime Numbers Between Two Intervals.
21.  Program to print stars Sequence1.
22. \*
23. \*\*
24. \*\*\*
25. \*\*\*\*
26. \*\*\*\*\*
27. Program to print stars Sequence2.
28. \*
29. \*\*
30. \*\*\*
31. \*\*\*\*
32. \*\*\*\*\*
33. Program to print star Sequences3.
34. \*
35. \*\*\*
36. \*\*\*\*\*
37. Program to print Sequence4.
38. \*
39. \*\*
40. \*\*\*
41. \*\*\*\*
42. \*\*\*\*\*
43. \*\*\*\*\*
44. \*\*\*\*
45. \*\*\*
46. \*\*
47. \*
48.  Sum of Numbers: Write a program that prompts the user for an integer n and then calculates the sum of all integers from 1 to n using a for or while loop. Also, calculate the sum of all even and odd numbers.
49.  Guess the Number Game: Create a simple game where the program picks a number (int number = 42;), and the user has to guess the number, receiving hints (higher or lower). Use a while loop to handle the game process. If the user guesses the number, stop the program and display the number of attempts made by the user.
50.  User Menu Interaction: Create a text menu that provides the user with several options (e.g., "1. Perform action 1", "2. Perform action 2," and so on). Use a while loop to continue the program until the user chooses the exit option (system("exit");).
51.  Program to display the series and find the sum of 1 + 3 + 5 + ... + n.
52.  Program to display the sum of the series 1 + 1/2 + 1/3 + ... + 1/n.
53.  Write a program to add the first seven terms of the following series using a for loop: 1/1! + 2/2! + 3/3! + ...
54.  C++ Program to Find GCD of Two Numbers.
55.  C++ Program to Find LCM of Two Numbers.
56.  C++ Program to Display Characters from A to Z Using Loop.
57.  C++ Program to Count Number of Digits in an Integer.
58.  C++ Program to Reverse a Number.
59.  C++ Program to Calculate the Power of a Number.
60.  C++ Program to Check Whether a Number is Palindrome or Not.
61.  C++ Program to Check Armstrong Number.
62.  C++ Program to Display Armstrong Number Between Two Intervals.
63.  C++ Program to Convert Binary Number to Decimal and vice-versa.
64.  C++ Program to Convert Octal Number to Decimal and vice-versa.
65.  C++ Program to Convert Binary Numbers to Octal and vice-versa.
66.  C++ Program to Reverse a Sentence Using Recursion.
67.  C++ Program to calculate the power using recursion.
68.  Write a program to calculate the area of a circle using functions.
69.  Simple Calculator Program: \*\*\*\*Create a program that acts as a basic calculator, allowing users to perform addition, subtraction, multiplication, and division. Use functions for each operation.
70.  Write a program to swap two values using functions.
71.  Write a program to convert time to minutes using functions. (input 3 variables namely hours, minutes, and seconds. Convert everything into minutes.)
72.  Write a program to sum the Fibonacci series up to n (input n). 1, 1, 2, 3, 5, 8, 13, …
73.  Function Overloading and Default Arguments: Build a program for calculating the area and perimeter of various geometric shapes (circle, rectangle, triangle, etc.) using separate functions for each shape. Overload functions for shapes with different parameters.
74.  Employee Payroll: \*\*\*\*Design a program that calculates employee payroll, including basic salary, overtime pay, and deductions. Use functions to compute each component.
75.  **Student Grade Tracker:** Create a program that allows teachers to enter student grades and calculate averages, find the highest and lowest scores, and display statistics.
76.  **Library Management System:** Create a simple library management system where you can store and manage a list of books using arrays. Ask the user to enter the book names. You should have the function display the book names. Create a void function. You should have the functionality to update the book name. To do this create another function. And pass index as argument.
77.  C++ Program to merge two arrays.
78.  C++ Program to search the value in the array and return its index, if the value is not found print “Item not found”.
79.  \*\***Number Sorting:** Write a program that reads a list of numbers into an array and sorts them in ascending or descending order using a sorting algorithm.
80.  **Matrix Operations:** Write a program for basic matrix operations, such as addition, subtraction, multiplication, and transposition.
81.  In a small company, there are five salesmen. Each salesman is supposed to sell three products. Write a program using a 2D array to print **(Input from user)**. The total sales by each salesman and Total sales of each item.
82.  C++ Program to Calculate Standard Deviation.
83.  C++ Program to Multiply Two Matrices by Passing the Matrix to a Function.
84.  C++ Program to Access Elements of an Array Using Pointer.
85.  Write a program that declares an integer variable, assigns a value to it, and then uses a pointer to print the value.
86.  Swap the values of two integer variables using pointers.
87.  Write a program that finds the sum of elements in an integer array using a pointer.
88.  Create a dynamic integer array and prompt the user for the array size. Fill the array with user input values.
89.  C++ Program Swap Numbers in Cyclic Order Using Call by Reference.
90.  C++ Program to Find Largest Number Using Dynamic Memory Allocation.
91.  C++ Program to Find the Frequency of Characters in a String.
92.  C++ Program to count the number of vowels, consonants, and so on.
93.  C++ Program to Remove all Characters in a String Except Alphabet.
94.  C++ Program to Find the Length of a String.
95.  C++ Program to Concatenate Two Strings.
96.  C++ Program to Copy String Without Using strcpy().
97.  C++ Program to Sort Elements in Lexicographical Order (Dictionary Order).
98.  C++ Program to Store Information(name, roll, and marks) of a Student Using Structure.
99.  C++ Program to Add Two Distances (in inch-feet) System Using Structures.
100.  C++ Program to Add Two Complex Numbers by Passing Structure to a Function.
101.  C++ Program to Calculate Difference Between Two Time Periods.
102.  C++ Program to Store Information of Students Using Structure.
103.  C++ Program to Store Information Using Structures with Dynamically Memory Allocation.
104.  C++ Program to Write a Sentence to a File.
105.  C++ Program to Read a Line From a File and Display it.
106.  C++ Program to Display its own Source Code as Output.
107.  C++ Programming Code To Create Pyramid and Pattern.
108.  [OOP] Define a class called **Car** with attributes like **model**, and **year**. Create an object of the **Car** class and set its attributes. Then, print out the car's details.
109.  [OOP] Redo the same program above using this->
110.  [OOP] Temperature converter. Write a class called **TemperatureConverter** with methods to convert between Celsius and Fahrenheit. From Celsius Kelvin. to Create an object of this class, and use it to convert a temperature from one scale to another.
111.  [OOP] Simple calculator. Create a class called **Calculator** that can perform basic arithmetic operations (addition, subtraction, multiplication, division). Create an object of the class and use it to perform some calculations.
112.  [OOP] Create a class **Rectangle** with attributes for its length and width. Implement a method to calculate the area of the rectangle. Create an object and compute the area for a specific rectangle.
113.  [OOP] Simple To-Do List: Design a basic to-do list application where users can add, remove, and display tasks. You can save tasks in an array.

Read a string from the user

Read your name and print it on the screen

Read a number from the user and print it on the screen

Read an option that hold yes or no from the user and perform specific action based on its value.

Read a char and store it then print it with its ascii code

Read two strings from the user and print them

Read three character from the user and print them

Read three numbers from the user and print them

Read examples:

Read your name and age then print them

Read your name and age , then print the length of your name.

Read your name and age then determine your age category based on your age

Program to read two numbers and add them

Subtract two numbers

Multiply two numbers

Divide two numbers

Get the remainder of two numbers

Certainly! Here's a list of operations typically associated with each of the entities mentioned in a bank system:

1. Customer:

- Create a new customer profile

- Update customer information (name, address, contact details)

- Close customer account

- View account balance and transaction history

- Apply for a new account or loan

1. Customer Profile:
   * Procedure: CreateNewCustomerProfile()
     + Prompt the user to enter customer details (name, contact information, etc.).
     + Generate a unique customer ID or assign an auto-incremented value.
     + Save the customer profile in the database or data storage.
2. Update Customer Information (Name, Address, Contact Details):
   * Procedure: UpdateCustomerInformation(customerID)
     + Retrieve the customer profile based on the provided customer ID.
     + Display the existing customer information.
     + Prompt the user to enter updated information (name, address, contact details).
     + Update the customer profile with the new information and save it.
3. Close Customer Account:
   * Procedure: CloseCustomerAccount(customerID)
     + Check if the customer has any outstanding balances or pending transactions.
     + If there are no outstanding balances or pending transactions:
       - Update the customer account status to "closed."
       - Transfer any remaining funds to the customer's specified account.
     + If there are outstanding balances or pending transactions:
       - Notify the customer about the outstanding balances or pending transactions.
       - Request the customer to settle the balances or complete the pending transactions.
       - Once resolved, proceed with closing the customer account.
4. View Account Balance and Transaction History:
   * Procedure: ViewAccountBalanceAndTransactionHistory(accountID)
     + Retrieve the account details based on the provided account ID.
     + Display the current account balance.
     + Retrieve and display the transaction history associated with the account (sorted by date, from the latest to the oldest).
5. Apply for a New Account or Loan:
   * Procedure: ApplyForNewAccountOrLoan(customerID)
     + Retrieve the customer profile based on the provided customer ID.
     + Prompt the customer to select the type of account or loan they want to apply for.
     + Gather necessary information for the selected account or loan type (e.g., account type, loan amount, repayment term).
     + Perform any required eligibility checks or validation.
     + Submit the application and notify the customer about the status (approval, rejection, or further review).
     + If approved, create a new account or loan record in the system and associate it with the customer.

2. Account:

- Open a new account for a customer

- Close an account

- Deposit funds into the account

- Withdraw funds from the account

- Transfer funds between accounts

- View account balance and transaction history

3. Transaction:

- Record a new transaction

- Retrieve transaction details

- Modify transaction details (e.g., amount, description)

- Cancel or reverse a transaction

- Generate transaction reports (daily, monthly, etc.)

4. Bank Employee:

- Create a new employee profile

- Update employee information (name, contact details, position)

- Assign employee roles and permissions

- View customer account details

- Assist customers with account-related inquiries

- Generate reports on employee performance

5. Loan:

- Create a new loan record for a customer

- Approve or reject loan applications

- Modify loan terms (e.g., interest rate, repayment schedule)

- Process loan repayments

- Generate loan statements and reports

6. Branch:

- Add a new branch location

- Update branch information (address, contact details, operating hours)

- View branch-specific reports (transactions, accounts, etc.)

- Assign employees to branches

- Manage branch resources (ATMs, cash, etc.)

7. ATM:

- Process cash withdrawals

- Accept cash deposits

- Display account balance and transaction history

- Provide account transfer functionality

- Generate ATM transaction receipts

- Perform card-related operations (e.g., card activation, PIN change)

8. Credit Card:

- Issue a new credit card to a customer

- Activate or deactivate a credit card

- Adjust credit limits

- Process credit card transactions

- Generate credit card statements

- Handle credit card disputes and fraud cases

9. Transaction Category:

- Create new transaction categories (e.g., deposits, withdrawals, fees)

- Modify existing transaction categories

- Categorize transactions based on predefined categories

- Generate reports based on transaction categories

10. Security:

- Authenticate user credentials

- Control access to system functionalities based on user roles

- Encrypt sensitive data (e.g., passwords, customer information)

- Monitor system logs and audit trails

- Implement security measures to prevent unauthorized access or fraud

- Generate security-related reports (e.g., login activity, access logs)

Certainly! Here's a list of operations typically associated with each entity in an inventory management system:

1. Product:

- Add a new product to the inventory

- Update product information (name, description, price, quantity)

- Remove a product from the inventory

- Search for products based on various criteria

- Adjust product quantities (e.g., restock, sell)

- Generate reports on product stock levels and performance

2. Supplier:

- Add a new supplier to the system

- Update supplier information (name, contact details, payment terms)

- Remove a supplier from the system

- Search for suppliers based on various criteria

- View supplier details and order history

- Generate reports on supplier performance and payments

3. Purchase Order:

- Create a new purchase order

- Add products to a purchase order

- Specify quantities and prices for products

- Send purchase orders to suppliers

- Track the status and progress of purchase orders

- Receive and process incoming shipments based on purchase orders

4. Warehouse:

- Add a new warehouse to the system

- Update warehouse information (name, address, capacity)

- Remove a warehouse from the system

- Assign products to specific locations within the warehouse

- Track inventory movements between different warehouse locations

- Generate reports on warehouse utilization and inventory distribution

5. Inventory Stock:

- View current stock levels of each product

- Receive new inventory stock and update quantities

- Adjust stock quantities due to sales or returns

- Track stock movements (e.g., transfers, adjustments)

- Monitor stock levels and set reorder points

- Generate reports on stock levels, expiration dates, and batch numbers

6. Sales Order:

- Create a new sales order for a customer

- Add products to a sales order

- Specify quantities and prices for products

- Calculate totals and apply discounts or taxes

- Generate invoices or receipts for sales orders

- Track the status and progress of sales orders

7. Customer:

- Add a new customer to the system

- Update customer information (name, contact details, addresses)

- Remove a customer from the system

- Search for customers based on various criteria

- View customer purchase history and order details

- Generate reports on customer activity and sales

8. Shipment:

- Create a new shipment record

- Associate a shipment with a sales order or return

- Specify shipping details (carrier, tracking number, delivery date)

- Update shipment status and track delivery progress

- Receive and process incoming shipments from suppliers

- Generate reports on shipment status and delivery performance

9. Return:

- Create a new return record for a customer

- Associate a return with a sales order or shipment

- Specify return reasons and return quantities

- Process returns and determine refund or replacement actions

- Update inventory stock quantities based on returned items

- Generate reports on return volumes and reasons

10. Inventory Report:

- Generate reports on product stock levels

- Analyze sales trends and product performance

- Review supplier performance and order history

- Monitor customer activity and purchase history

- Generate financial reports related to inventory and sales

- Provide analytics and insights for decision-making purposes

These operations represent common functionalities associated with each entity in an inventory management system. The actual set of operations may vary depending on the specific requirements and design of the system.

These operations represent common functionalities associated with each entity in a bank system. The actual set of operations may vary depending on the specific requirements and design of the bank system.

Write a program to convert between the computer memory units such as bytes, kilobyte, megabyte, and so on.

Write a program to convert the year, days, hours, minutes into seconds and vise versa