**C Assignments**

**Level: Easy**

1. Write a C Program to find the greatest of 3 numbers.
2. Implement a basic calculator (+, -,\*, /, %) using C Programming.
3. Write a C Program to find the sum up to N numbers.

Example: Enter the number N: 5

Sum up to 5 is: 15

Enter the number N: 3

Sum up to 3 is: 6

1. Write a C Program which adds all numbers from 1 to N, except those which are divisible by 5.

Example: Enter the number N: 5

Sum up to 5 is: 10

Enter the number N: 15

Sum up to 15 is: 90

1. Write a C program to store n elements in an array and print them.
2. Write a C program to store n elements in an array and print those using pointers.
3. Write a C program to read n number of values in an array and display them in reverse order.
4. Write a C program to find the sum of all elements of the array.
5. Write a C program to copy the elements of one array into another array.
6. Write a C program to count the total number of duplicate elements in an array.
7. Write a C program to print all unique elements in an array.
8. Write a C Program to sort a given array in:
9. Ascending Order
10. Descending Order
11. Implement a function to count the frequency of each element of an array using pointers.
12. Implement a function to find the maximum and minimum elements in an array using pointers.
13. Implement a function to separate odd and even integers of an array into separate arrays using pointers.
14. Implement a function to find the second largest element in an array using pointers.
15. Implement a function to find the second smallest element in an array using pointers.
16. Write a C Program to read a 2D array of size 3x3 and print the matrix.
17. Write a C program for adding two matrices of the same size.
18. Write a C program for the subtraction of two matrices.
19. Write a C program to find the sum of rows and columns of a matrix.
20. Write a C program to input a string and print it.
21. Write a C program to find the length of a string without using library functions.
22. Implement a function to copy one string to another string using pointers.
23. Implement a function to count the total number of vowels or consonants in a string.
24. Implement a function to find the frequency of characters.

Example:

Input the string: This is a test string

Input the character to find frequency: i

Expected Output: The frequency of 'i' is: 3

1. Implement a function to find the largest and smallest words in a string using pointers.

Example:

Input the string: It is a string with the smallest and largest word.

Expected Output:

The largest word is: 'smallest'

The smallest word is: 'a'

1. Implement a function to reverse a string using pointers.
2. Implement a function to check if a string is a palindrome using pointers.
3. Create a structure called "Student" with members' names, ages, and total marks. Write a C program to input N students' data and display their information.
4. Create a structure named Book to store book details like title, author, and price. Write a C program to input details for N books, find the most expensive and the lowest priced books, and display their information.
5. Write a C program to create and store information in a text file.
6. Write a C program to read an existing file.
7. Write a C program to find the number of lines in a text file.
8. Write a C program to count the number of words and characters in a file.
9. Write a C program to delete a specific line from a file.
10. Write a C program to copy the content from one file to another file.

**Level: Medium**

1. Ramesh hires a labourer to paint the wall of his home. The length and width of the wall, the paint required per square foot, the time required per square foot, and the labour charge per square foot are stored in a file.

Your task is to read this data from the file and then calculate and display it on the console:

Total area of the wall.

Total paint required (in mL).

Total time required (Hrs or Minutes).

Total labour cost (in ₹).

1. A person wants to track their daily expenses for a week.

The user enters daily expenses for 7 days. The program should calculate and display the total and average expenses. Find the day with the highest and lowest expense. If total expenses exceed a certain budget (e.g., ₹10,000), display: "You have exceeded your budget!”.

1. In a cricket tournament, there are 5 batsmen each from the Indian team and the Australian team. The Media Manager needs to display the list of batsmen from both teams in descending order of their batting average and display batsmen with the same batting average from both teams.

(**Hint:** Use **Structures** - to store batsman details and arrays - to store team data)

Example: Enter details of Indian players:

Name of Batsman1: Virat Kohli

Total matches played: 5

Total runs scored: 355

Name of Batsman2: Hardik Pandya

Total matches played: 4

Total runs scored: 250

Name of Batsman3: Rohit Sharma

Total matches played: 5

Total runs scored: 600

Name of Batsman4: Surykumar Yadav

Total matches played: 4

Total runs scored: 350

Name of Batsman5: Abhishek Sharma

Total matches played: 4

Total runs scored: 377

Enter details of Australian players:

Name of Batsman1: Travis Head

Total matches played: 5

Total runs scored: 800

Name of Batsman2: Gleen Maxwell

Total matches played: 5

Total runs scored: 250

Name of Batsman3: Steve Smith

Total matches played: 4

Total runs scored: 410

Name of Batsman4: Cameron Green

Total matches played: 5

Total runs scored: 368

Name of Batsman5: Mitchell Marsh

Total matches played: 5

Total runs scored: 280

Output:

Table 1 Team: India

Batsman Name Matches Played Runs Scored Batting Average

Rohit Sharma 5 600 120.00

Abhishek Sharma 4 377 94.25

Virat Kohli 5 355 71.00

Suryakumar Yadav 4 350 87.50

Hardik Pandya 4 250 62.50

Table 2 Team: Australia

Batsman Name Matches Played Runs Scored Batting Average

Travis Head 5 800 160.00

Steve Smith 4 410 102.50

Cameron Green 5 368 73.60

Mitchell Marsh 5 280 56.00

Glenn Maxwell 5 250 50.00

Table 3: Batsmen with the Same Batting Average

Indian Team Batsman Australian Team Batsman Batting Average

None None No same batting average found!

Note: If any batsmen are found with the same batting average, then Table 3 should look as follows:

Example: Manish Pande and David Warner have the same batting average of 60.00 then

Table 3: Batsmen with the Same Batting Average

Indian Team Batsman Australian Team Batsman Batting Average

Manish Pande David Warner 60.00

1. A power distribution company wants to develop a C program to generate monthly electricity bills for its customers based on their electricity consumption.

Requirements:

User Input:

Customer Name

Customer ID (Validate whether customer ID is 6 digits or not)

Billing Month (e.g., January, February, etc.)

Previous Meter Reading (kWh)

Current Meter Reading (kWh)

Is the payment late? (1 for Yes, 0 for No):

Billing Calculation:

The bill is calculated based on the following slab rates:

|  |  |
| --- | --- |
| Units Consumed (kWh) | Rate per Unit (₹) |
| 0 - 100 | 3.50 |
| 101 - 300 | 5.00 |
| 301 - 500 | 7.50 |
| Above 500 | 10.00 |

A fixed charge of ₹100 is added to every bill.

Late Payment Surcharge (LPS): If the bill is paid after the due date, a 5% surcharge is added.

Example:

Input:

Enter Customer Name: Rahul Sharma

Enter Customer ID: 102345

Enter Billing Month: January

Enter Previous Meter Reading (kWh): 1250

Enter Current Meter Reading (kWh): 1400

Is payment late? (Enter 1 for Yes, 0 for No): 1

Output:

-------------------------------------------------

ELECTRICITY BILL - JANUARY 2025

-------------------------------------------------

Customer Name: Rahul Sharma

Customer ID : 102345

Billing Month: January

Previous Reading: 1250 kWh

Current Reading: 1400 kWh

Units Consumed: 150 kWh

-------------------------------------------------

Charges Breakdown:

- First 100 Units: 100 × 3.50 = ₹350.00

- Next 50 Units: 50 × 5.00 = ₹250.00

- Fixed Charge: ₹100.00

-------------------------------------------------

Total Bill Amount Before Due Date: ₹700.00

Late Payment Surcharge 5% (Applicable or Not Applicable): Applicable

Late Payment Fee (5% of ₹700): ₹35.00

-------------------------------------------------

Final Bill Amount: ₹735.00

-------------------------------------------------

Thank you for using our service!

**Level: Difficult**

1. The store manager of the Supermarket, Mr Sharma, wants to implement an automatic billing system that can quickly calculate the total cost of purchased groceries, including GST, and update stock levels in real time.

Here are the constraints provided by Mr Sharma:

A file (grocery\_data.txt) stores grocery details:

Item Name

Rate per Kg (₹)

GST Percentage

Available Stock (Kg)

The user enters the required groceries and their quantities.

The system searches for items in the file and:

If available, calculate the total cost (including GST).

If the requested quantity exceeds stock, notify the user: "Only X Kg available. Would you like to proceed with X Kg?"

If out of stock, notify the user: "Sorry, [Item] is out of stock."

The system generates a receipt displaying:

Item-wise cost (Base Price + GST).

Total payable amount.

Stock is updated in the file after the transaction.

Example:

Input:

Welcome to Supermarket Self-Checkout!

Enter the number of items you want to purchase: 3

Enter item name: Rice

Enter quantity (Kg): 5

Enter item name: Onion

Enter quantity (Kg): 5

Only 3 Kg is available. Would you like to proceed with 3 Kg? (Yes/No): Yes

Enter item name: Wheat

Enter quantity (Kg): 2

---------------------Supermarket Digital Bill ------------------------

Item Quantity Price GST Final Price

Rice 5Kg ₹250 5% ₹262.50

Onion 3Kg ₹75 5% ₹78.75

Wheat 2Kg ₹500 12% ₹560.00

---------------------------------------------------------------------------------

Total Amount Payable: ₹901.25

Thank you for shopping with Supermarket!

---------------------------------------------------------------------------------

Implementation Guidelines:

* Use structures to store grocery details.
* Implement error handling (e.g., item not found, insufficient stock).
* Ensure the program updates stock in the file after billing.

1. The Apana Bank wants to develop an Automated Banking Management System to provide customers with secure and efficient banking services. The system will allow users to perform essential banking operations such as account creation, deposits, withdrawals, balance inquiry, and transaction history tracking.

The Apana Bank requires the following features:

System Features:

1. Account Creation: The system will first display the available account types and benefits.

The customer selects the account type.

The system asks for Name, Initial Deposit, and Contact Details.

A unique Account Number is generated and stored in a file (accounts.txt).

2. Deposit Money: Customers can deposit money into their account. The system updates the balance and logs the transaction in a file (transactions.txt).

3. Withdraw Money: Customers can withdraw money if sufficient balance is available. The system updates the balance and logs the withdrawal.

4. Balance Inquiry: Customers can check their current account balance.

5. Transaction History: The system maintains a record of all deposits and withdrawals for each account and retrieves transaction history when requested.

6. Fixed Deposit (FD) Management: Customers can open an FD by selecting an amount and duration. The system calculates and displays the maturity amount based on the FD interest rate.

Example:

Input:

Welcome to Apana Bank!

1. Open New Account

2. Deposit Money

3. Withdraw Money

4. Check Balance

5. View Transaction History

6. Open Fixed Deposit

7. Exit

Enter your choice: 1

--- Account Types Available ---

1. Savings Account

- Earns interest (6% annually)

- Minimum balance required: ₹1000

- Withdrawal limit: ₹50,000 per day

2. Current Account

- No interest

- Minimum balance required: ₹5000

- No withdrawal limit

- Ideal for businesses

3. Fixed Deposit (FD)

- Higher interest rates (8%)

- Cannot withdraw before maturity

Enter the type of account you want to open (1-3): 1

Enter your name: Ramesh

Enter initial deposit amount: ₹5000

Enter contact number: 9876543210

Output:

Account created successfully!

Your Account Number: 1001

------------------------------

Input:

Enter your choice: 3

Enter Account Number: 1001

Enter Amount to Withdraw: ₹12,000

Output:

Insufficient Balance! Available Balance: ₹10,000

Input:

Enter your choice: 5

Enter Account Number: 1001

Output:

------ Transaction History --------

Deposit: ₹2000 on 12-02-2025

Withdraw: ₹1000 on 13-02-2025

Deposit: ₹5000 on 14-02-2025

------------------------------

Implementation Guidelines:

* Use structures to store account details and transactions.
* Use file handling to maintain account and transaction records.
* Ensure proper balance validation before withdrawals.
* Implement interest calculation for Savings Accounts & Fixed Deposits.
* Update account details after each transaction.

1. Arjun is an enthusiastic puzzle solver who loves playing Sudoku. However, he often gets stuck on puzzles and needs a Sudoku Solver to help him find a solution. He decides to develop a C program that can take an incomplete Sudoku puzzle as input and generate a valid solution. Help Arjun to develop a Sudoku Solver program using C programming.

[Sudoku is a popular number puzzle game played on a 9×9 grid divided into nine 3×3 subgrids. The goal is to fill the grid so that each row, each column, and each 3×3 subgrid contains all the digits from 1 to 9 exactly once.

Some numbers are pre-filled as clues, and the player must deduce the missing numbers based on these clues.]

Example:

Input:

Enter the incomplete Sudoku:

5 3 0 0 7 0 0 0 0

6 0 0 1 9 5 0 0 0

0 9 8 0 0 0 0 6 0

8 0 0 0 6 0 0 0 3

4 0 0 8 0 3 0 0 1

7 0 0 0 2 0 0 0 6

0 6 0 0 0 0 2 8 0

0 0 0 4 1 9 0 0 5

0 0 0 0 8 0 0 7 9

Output:

The solution for the given Sudoku is:

5 3 4 6 7 8 9 1 2

6 7 2 1 9 5 3 4 8

1 9 8 3 4 2 5 6 7

8 5 9 7 6 1 4 2 3

4 2 6 8 5 3 7 9 1

7 1 3 9 2 4 8 5 6

9 6 1 5 3 7 2 8 4

2 8 7 4 1 9 6 3 5

3 4 5 2 8 6 1 7 9