tcg crest
Inventing Harmonious Future

Institute for Advancing Intelligence, TCG CREST
(TCG Centres for Research and Education in Science and Technology)

Introduction to Programming and Data Structures
Ph.D. Coursework: First year, First Semester (Session: 2024-25)

Warm-up problem set

Maximum Marks: 00 Submission Deadline: **2024-Aug-19**

Problem number #AP0001:

Matrix Multiplication: Write a C program to multiply two matrices. User may request for up to 10 multiplications.

Problem number #AP0002:

ATM Withdrawal: Write a program that simulates an ATM withdrawal. The program should perform the following tasks:

1. Prompt the user to enter the amount they wish to withdraw.
2. Check if the amount is a multiple of 100 (since the ATM only dispenses 100, 200, 500, and 2000 denomination notes).
3. If the amount is not a multiple of 100, print: `"Invalid amount. Please enter an amount in multiples of 100."`
4. If the amount is valid, check if the user's account balance is sufficient for the withdrawal.
5. If the balance is insufficient, print: `"Insufficient funds."`
6. If the balance is sufficient, deduct the amount from the balance and print the remaining balance.
7. Ensure the program handles invalid input (e.g., negative numbers, non-numeric input) by displaying an error message and prompting the user to enter the amount again.

Examples:

- **Input:** Enter the amount you wish to withdraw: 250
  Output: `Invalid amount. Please enter an amount in multiples of 100.`
- **Input:** Enter the amount you wish to withdraw: 500
  Output: `Your remaining balance is Rs.1500.`

Note: Set a random balance between 0 to 20000 in the beginning. `rand()` function from <`stdlib.h`> may be used.

Problem number #AP0003:

Number Guessing Game: Write a program that implements a number guessing game. The program should:

1. Generate a random number between 1 and 100 (inclusive).
2. Allow the user to guess the number, giving them up to 10 attempts to guess correctly.
3. After each guess:
   (a) If the guess is lower than the random number, print: `"Too low!  Try again."`
   (b) If the guess is higher than the random number, print: `"Too high!  Try again."`
   (c) If the guess is correct, print: `"Congratulations!  You've guessed the number."` and end the game immediately using a `break` statement.
4. If the user inputs a number outside the range of 1 to 100, display an error message and use `continue` to skip that attempt without counting it as one of the 10 allowed attempts.
5. If the user fails to guess the number within 10 attempts, end the game and reveal the correct number.
6. Allow the user to exit the game at any time by entering a special character or word (e.g., `'q'` for quit).

Example I/O:

```
Welcome to the Number Guessing Game!
You have 10 attempts to guess the number between 1 and 100.
Enter your guess: 50
Too high! Try again.

Enter your guess: 25
Too low! Try again.

Enter your guess: 40
Congratulations! You've guessed the number.

Enter your guess: 150
Invalid input! Enter a number between 1 and 100.

Enter your guess: q
You've chosen to quit the game. Goodbye!
```