Data Science in Practice (DSE 315/615)

Dr. Bhavna Rajasekaran

August 9, 2023

Assignment

1. Print the decimal numbers 84 and 101 in binary, hexadecimal, and octal using python. Find their bit-wise complementary outputs. Perform 2-bit right shift operator on 84 and 2-bit left shift operator on 101. On the results obtained, **perform bit-wise OR and bit-wise AND operations**.

2. Print values from 1 to 20 while skipping values divisible by three using while loop

3. Modify code (prog9.py) for user input integer values & ~~amp;~~ check the outputs

4. Use a condition & ~~amp;~~ decrement ($i = i - 1$) the value of 'i' to print 'Hello world' 20 times; what will be the value of 'i' to start with? Also, print the value of 'i' at each iteration and 'Hello world.' E.g., Hello World 2 ($i = 2$).

5. Write a program that simulates a basic ATM transaction. Ask the user for their account balance and the amount they want to withdraw. If the balance is sufficient, subtract the withdrawal amount and print the new balance. If the balance is insufficient, print "Insufficient funds."

6. Create a program that generates a random math problem (e.g., addition, subtraction, multiplication) and asks the user to solve it. Keep track of the user's score and provide a final score at the end.

7. Develop a program that calculates the square root of a given positive number using the Newton-Raphson method with a while loop for iterative approximation.

8. Develop a program that simulates a basic calculator with the following operations: addition, subtraction, multiplication, division, and exponentiation. Allow the user to choose an operation and provide the necessary inputs for the chosen operation.

9. Create a program that simulates a simple quiz with multiple-choice questions. Ask a series of questions and give the user a score based on their answers. Include a mix of topics and varying levels of difficulty.

10. Create a program that generates a list of random numbers and sorts them using the bubble sort algorithm within a while loop. Keep track of the number of iterations needed for sorting.

11. Develop a program that generates a random number between 1 and 100. Allow the user to guess the number, providing hints such as "higher" or "lower" after each incorrect guess. Continue until the user guesses correctly or reaches a maximum number of attempts.