PLACEMENTDOST

Python Assessment questions

Note:

Please be advised that these assessment inquiries are designed to accommodate interns with diverse skill levels, ranging from novices to seasoned analysts. Should you encounter any challenging questions, you are encouraged to seek solutions independently or reach out to us for assistance at intern@placementdost.com. Best wishes for success in completing the assessment!

Do any 15 out of these 20 problem statements and make a zip file of source code

Section A: Python Basics

1. Write a Python program to swap the values of two variables without using a temporary variable.

2. Create a function that checks if a given number is prime.

3. Implement a program to find the factorial of a given number using recursion.

4. Write a Python script that reads a text file, counts the occurrences of each word, and prints the results.

5. Define a class representing a basic calculator with methods for addition, subtraction, multiplication, and division.

6. Create a Python program to generate a list of the first 20 Fibonacci numbers.

7. Write a function that accepts a list of integers and returns a new list containing only the even numbers.

8. Implement a program to calculate the average of a list of numbers without using the built-in `sum` and `len` functions.

9. Develop a Python script that takes a user's input and prints whether it is a palindrome or not.

10. Create a program that uses a dictionary to count the frequency of each word in a given text.

Section B: Intermediate Python

11. Implement a function that reverses a string without using slicing or reverse functions.

12. Write a program to find the common elements between two lists.

13. Develop a Python script that reads data from a CSV file, performs data cleaning, and prints a summary report.

14. Create a context manager that measures the execution time of a code block.

15. Write a generator function that yields prime numbers indefinitely.

16. Implement a program that counts the number of vowels in a given string using a dictionary.

17. Develop a decorator that logs the arguments and return value of a function.

18. Create a program that handles exceptions when dividing two numbers, prompting the user to enter valid inputs.

19. Write a Python script to download an image from a URL using the `requests` library.

20. Implement a class-based stack with methods for push, pop, and checking if the stack is empty.