PART A: Short Questions (Basic Concepts and Small Programs)

1. Write a Python program to swap two numbers without using a temporary variable.
2. Write a Python program to print the Fibonacci sequence up to a user-provided number n.
3. Write a Python program to find the largest number in a list without using the max() function.
4. Write a Python program to check if a given year is a leap year.
5. Write a Python program to reverse a string without using slicing.

PART B: Programming and Problem-Solving (Moderate Difficulty)

1. Write a Python program to find all prime numbers within a user-specified range.
2. Write a Python program to count the frequency of each character in a string.
3. Create a Python program to calculate the sum of squares of all even numbers in a list.
4. Write a Python program to merge two dictionaries and handle duplicate keys by summing their values.
5. Write a Python program to check if two strings are anagrams of each other.
6. Develop a Python program to rotate a list to the right by k positions.
7. Write a Python program to calculate the GCD (Greatest Common Divisor) of two numbers using recursion.
8. Create a program that generates a multiplication table for a given number up to a specified range.
9. Write a Python program to flatten a nested list into a single list.
10. Implement a program that removes duplicate elements from a list while maintaining the original order.

PART C: Advanced Questions (Challenging)

1. Write a Python program to implement a basic text-based calculator that supports addition, subtraction, multiplication, division, and exponentiation.
2. Create a program that validates a user-provided email address using regular expressions.
3. Develop a Python program to calculate the sum of all numbers in a string (e.g., input: a1b2c3, output: 6).
4. Write a Python program to simulate a basic banking system with the following features:
   - Deposit money
   - Withdraw money
   - View balance
5. Create a program to implement matrix multiplication without using built-in functions.
6. Write a Python program to generate all possible subsets (the power set) of a given list.
7. Implement a Python program to solve the Tower of Hanoi problem for a user-specified number of disks.
8. Create a Python program that reads a text file and prints the top 3 most frequent words in it.

9. Write a Python program to encrypt a string using Vigenère cipher and decrypt it back.

10. Develop a Python program that analyzes a list of student marks and generates a report with the following:
    - Average marks
    - Highest and lowest marks
    - Number of students passing (marks >= 50)

11. Write a program to:

- Analyze sales data from the following dataset:

```
sales_data = {
    'Product A': [100, 200, 150],
    'Product B': [300, 400, 250],
    'Product C': [50, 80, 100]
    }
```

- Calculate the total and average sales for each product.
- Identify the product with the highest average sales.

12.

```
students = {
   'John': {'courses': 5, 'assignments': [{'marks': 95, 'feedbacks': 15}, {'marks': 85, 'feedbacks': 10}]},
   'Jane': {'courses': 6, 'assignments': [{'marks': 90, 'feedbacks': 25}, {'marks': 80, 'feedbacks': 20}]},
   'Smith': {'courses': 4, 'assignments': [{'marks': 100, 'feedbacks': 30}, {'marks': 95, 'feedbacks': 15}]}
}
```

Analyze the given dataset and implement the following features:

- Find the transaction with the highest value (amount).
- Identify the store with the most customers.