CODEFEST

Task 1

Scenario: Wardriving prevention

Wardriving is the act of systematically scanning and mapping the availability of wireless networks, typically by driving or walking around urban areas with a Wi-Fi-enabled device. Individuals engaged in wardriving may use specialized software to identify and document open or vulnerable Wi-Fi networks for various purposes, including security testing or unauthorized access.

You are tasked with enhancing the security of a Wi-Fi network by implementing a robust password validation function. The goal is to prevent unauthorized access to the network, particularly by wardriving attempts. Your password validation function should consider the following criteria:

1. Length Requirement: The Wi-Fi password must be at least 12 characters long.

2. Character Set: The password must include a combination of characters from at least three of the following sets:

    Uppercase letters (A-Z)
    Lowercase letters (a-z)
    Digits (0-9)
    Special characters (e.g., @, #, $, %, ^, &, *)

3. No Sequential Characters: The password cannot contain a sequence of three or more sequential characters (e.g., "abc," "123," "XYZ").

CODEFEST

4. ***BONUS -*** No Dictionary Words: The password should not contain common dictionary words, to prevent dictionary attacks. You may use an online dictionary API (assume it's available) to check for common words.

Write a function named **`validate_wifi_password()`** that takes a Wi-Fi password as input and returns **`True`** if the password meets the criteria, and **`False`** otherwise. Provide both the function implementation and an example of using the function to test various Wi-Fi passwords.

Note: An object-oriented approach is suggested, which in turn might give you bonus marks if you implement it in an understandable manner. Feel free to use any programming language you are comfortable with.

Task 2

Scenario: Online Bookstore Inventory Management

You are tasked with developing a program for an online bookstore to manage its inventory. The bookstore sells a variety of books, each identified by a unique ISBN (International Standard Book Number). The inventory is stored in a file called "inventory.txt," where each line represents a book with the following format:

ISBN,Title,Author,Quantity,Price

Here's a sample entry:

978-0-13-475416-9,Introduction to Algorithms,Thomas H. Cormen,50,79.99

The program needs to perform the following tasks:

CODEFEST

1. Read and Display Inventory:

   Read the "inventory.txt" file and display the current inventory with ISBN, Title, Author, Quantity, and Price for each book.
   Add New Books:

2. Allow the user to add new books to the inventory.

   The program should prompt the user for the ISBN, Title, Author, Quantity, and Price of the new book. Update the "inventory.txt" file with the new information.

3. Update Inventory:

   Allow the user to update the quantity of existing books in the inventory. The program should prompt the user for the ISBN and the new quantity. Update the "inventory.txt" file accordingly.

4. Search and Display Book Information:

   Allow the user to search for a book by ISBN and display its information, including Title, Author, Quantity, and Price.

5. Calculate Total Inventory Value:

   Calculate and display the total value of the current inventory based on the quantity and price of each book.

6. Generate Sales Report:

   Create a sales report that lists the books that have been sold. The program should prompt the user for the ISBN and quantity sold for each transaction. Update the inventory and create a sales report file called "sales_report.txt" with the transaction details.

7. Data Validation:

   Implement data validation to ensure that the user enters valid numerical values for quantities and prices. Display appropriate error messages if invalid data is entered.

8. Menu System:

   Implement a menu system that allows the user to choose from the above tasks. Continue running the program until the user chooses to exit.

CODEFEST

9. Error Handling:

    Implement error handling to gracefully handle situations such as a missing "inventory.txt" file or incorrect data format within the file.

10. Code Documentation:

    Provide clear and concise comments in your code to explain the purpose and functionality of each section.

Feel free to use any programming language of your choice, but ensure that your code is well-organized, follows best practices, and includes appropriate error handling.

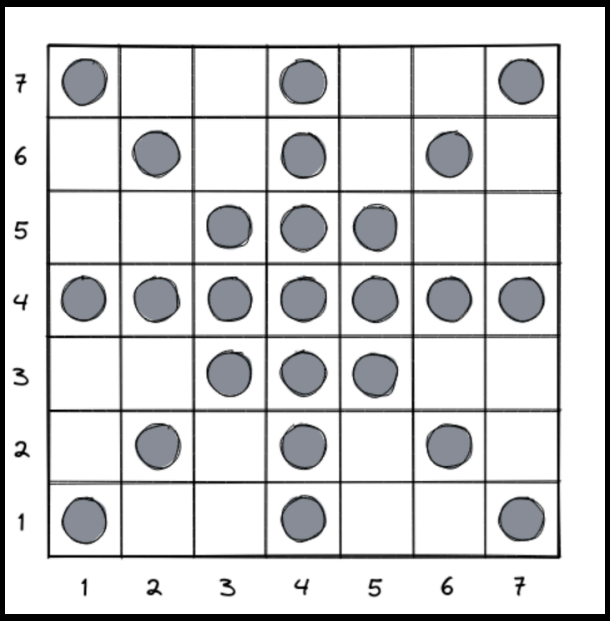
Task 3

Scenario: Calculating the number of dots on a grid

Nuzayer loves to do art. He has a graph, which has n rows and m columns.

He creates new art every day. Each piece of art costs a certain amount of ink to create. So he buys ink from the market accordingly. Today Nuzayer wants to draw a star on the graph and Nuzayer wants the area of his star to be maximum. Nuzayer and you are good friends. As such, Nuzayer wants to know from you, how many boxes of the graph should be filled with ink to make a star.

A star in a 7∗7 grid will look like the image below:



CODEFEST

Each arm from the center of the star is equal in length.

Input

The first line will contain 2 integers n and m ($1 \le n,m \le 10^9$) — the number of rows and columns, respectively. It is guaranteed that both n and m will be odd.

Output

Print how many boxes to color.

Samples

Input	Output
3 3	9
5 7	17
9 21	33

CODEFEST

Task 4

Scenario: Generating a pattern

Design a function named `generate_triangle_pattern()` that takes an integer n as input and prints a right-angled triangle pattern where each row contains multiples of a unique integer. The pattern should start with the multiples of 2 and increase by 2 for each subsequent row. The function should produce the following pattern for n = 5:

```
2
4 8
6 12 18
8 16 24 32
10 20 30 40 50
```

Implement the `generate_triangle_pattern()` function and provide an example demonstrating its usage for different values of n. Explain the algorithmic approach you use to create the triangular pattern with multiples within comments or docstrings and use a programming language of your choice for the implementation.