Lab-02 Task

Task-01

Declare a 2D array of integers using ONLY POINTERS. The row and column dimensions will be taken as user input. Then, loop through this input array using ONLY POINTER SYNTAX (not arr[i][j]). You must print on the screen the total sum of ALL ROWS and ALL

COLUMNS separately.

Task-02

Write a C++ program with a function findIntersection that takes two integer arrays, arr1 and arr2,along with their sizes. The function dynamically allocates and returns an array containing unique elements common to both arrays. The size of the resulting array is stored in the variable pointed to by resultSize. If no intersection, return NULL.

Task-03

Develop a C++ program that dynamically allocates memory for a 2D array of dimensions M x N based on userinput. The program should perform the following actions: Utilize dynamic memory allocation and pointers for 2D arrays to extend the size of the initial input array. The expansion of the array should be determined by the corners of the input array. The top-left corner designates the number of rows to add above the input array. The top-right corner designates the number of colums to add to the right of the input array. The bottom-left corner designates the number of rows to add below the input array. All newly added rows and columns must be populated with zeros. (Ignore the bottom right corner of the input matrix.)

Input Output

2 4

2 3 4 1

1 6 7 2

Output

0 0 0 0 0

0 0 0 0 0

2 3 41 0

1 6 7 2 0

0 0 0 0 0

Task-04

Develop a C++ program to assist a store in managing its inventory dynamically. Allow the user to input the desired size of the inventory array, and then generate random quantities of product items to fill the array.

Your program should address the following tasks using pointer notation:

1. Calculate the Average Stock Level: Find the mean (average) stock level of all the products in the inventory.

2. Identify Critical Products: Determine and count the products in the inventory that are critically low (less than the average) in stock.

3. Discover Top-Selling Product: Call a method named findTopSellingProduct to identify and print the product that sells most frequently. In the case of ties, highlight only one top-selling product.

4. Find Second Best-Selling Product: Implement a function, findSecondBestSeller, to identify and return the second best-selling product based on sales data.

5. Sort Products by Popularity: Create a function named sortByPopularity to sort the products in descending order based on their sales volume.

By addressing these tasks, your program can provide valuable insights for inventory management, helping the store optimize stock levels, identify crucial items, and enhance overall product performance.

Task-05 Bonus

There is famous Conjecture in mathematics that says that every even integer greater than 2 has the property that it is the sum of two prime numbers (at-least 1 pair, there can be more than 1). Like, For the even number 8, the Goldbach pair is (3, 5) because 3 + 5 = 8. For 12, the Goldbach pair is (5, 7) because 5 + 7 = 12. Similarly, for 28, one Goldbach pair is (5, 23) because 5 + 23 = 28, and also (11, 17) because11 + 17 = 28. Computers have been used extensively to test this conjecture. No counterexample has been

found to disprove this conjecture till date. Give implementation to the function definition below that finds all Goldbach pairs for all the n even numbers passed as pointer argument.

The function returns a pointer to memory location of an 2D-Array containing pairs of goldbach pairs for the integer numbers at even_numbers with last pair being -1 -1.

int** goldbach_pairs(int* even_numbers, int n);

Use the above function in a C program that takes list of even numbers from the user until he types -1. After which the above function should be used and the pairs returned through the memory location should be printed.

Input Sample

8 12 28 50 -1

Output Sample

(3, 5) (5, 7) (5, 23) (11, 17) (5, 45) (7, 43) (13, 37)

(19, 31) (-1,-1)