Programming Fundamental

Programming Day - Week 09

Introduction

Welcome to your favorite day of the week which is programming day🎉. This week, we shall work together to learn and implement new programming concepts.

Let's do some coding.

Task 01(CP):

Suppose you have the following fruits in one array and their prices in corresponding price array.
fruit = ["peach", "apple", "guava", "watermelon"];
price = [60, 70, 40, 30]
(Hint: the price and fruit name are on the same indexes but in different arrays, (these are called Parallel Arrays))

Ask the user to enter the name of the fruit as well as the quantity in kgs of the fruit. Then search the array for the fruit name and calculate the price of the fruit after searching from the price array then display the total bill.

Task 02(CP):

Suppose a Cinema displays 5 movies. Price of Each movie ticket is 500. Write a program in which you have 5 movie names stored in the array.
For Example: movies[5] = {Gladiator, StarWars, Terminator, TakingLives, TombRider}

Take 1 movie name as input from the user and if the movie is stored on an odd index of the array then give 5% discount on the movie ticket otherwise give 10% discount.

Task 03(CP):

Write a C++ program that is given a string as input, it returns true if its length is even and false if the length is odd.

Test Cases

Input	Output	Explanation
Enter a String: apples	true	// The word "apples" has 6 characters.

		// 6 is an even number, so the program outputs true.
Enter a String: pears	false	// "pears" has 5 letters, and 5 is odd. // Therefore the program outputs false.
Enter a String: cherry	false	

Note: Make all the arrays and their indexes global (only the arrays and the variable containing their indexes) so that you can use those in any function.

Task 04(CP):

Write a program that takes an array of numbers as input from the user and then prints "Boom!" if the digit 7 appears in the array. Otherwise, print "there is no 7 in the array".

Test Cases:

[1, 2, 3, 4, 5, 6, 7] ➞ "Boom!"
// 7 contains the number seven.
[8, 6, 33, 100] ➞ "there is no 7 in the array"
// None of the items contain 7 within them.
[2, 55, 60, 97, 86] ➞ "Boom!"
// 97 contains the number seven

Task 05(CP):

Create a program that checks in an array (slot machine outcome) and prints true if all elements in the array are identical, and false otherwise. The array will contain 4 elements.

Test Cases:

["@", "@", "@", "@"] ➞ true
["abc", "abc", "abc", "abc"] ➞ true
["SS", "SS", "SS", "SS"] ➞ true
["&&", "&", "&&&", "&&&&"] ➞ false
["SS", "SS", "SS", "Ss"] ➞ false

Task 06(CP):

Write a C++ program that performs an even-odd transform to an array, n times. One even-odd transformation is

a. Adds two (+2) to each odd integer.
b. Subtracts two (-2) to each even integer.

Test Cases:

Input	Output	Explanation
Enter the array: [3, 4, 9] Enter number of times even-odd transformation need to be done: 3	[9, -2, 15]	// Since even-odd transformation needs to be applied 3 times [3, 4, 9] becomes => [5, 2, 11] => [7, 0, 13] => [9, -2, 15]
Enter the array: [0, 0, 0] Enter number of times even-odd transformation need to be done: 10	[-20, -20, -20]	
Enter the array: [1, 2, 3] Enter number of times even-odd transformation need to be done: 1	[3, 0, 5]	

Task 07(CP):

Write a C++ program that is Given two strings, find the number of common characters between them and then prints that count.

Test Cases: For s1 = "aabcc" and s2 = "adcaa", the output should be 3.
Strings have 3 common characters; 2 "a"s and 1 "c".

Task 08(CP):

When coloring a striped pattern, you may start by coloring each square sequentially, meaning you spend time needing to switch coloring pencils.

Create a program where given an array of colors cols, it prints how long it takes to color the whole pattern. Note the following times:

- It takes 1 second to switch between pencils.
- It takes 2 seconds to color a square.

See the example below for clarification.

Test Cases:

Input	Output	Explantion

["Red", "Blue", "Red", "Blue", "Red"]	14	// There are 5 colors so it takes 2 seconds to color each one (2 x 5 = 10).
		// You need to switch the pencils 4 times and it takes 1 second to switch (1 x 4 = 4).
		// 10 + 4 = 14

["Blue"]	2	
["Red", "Yellow", "Green", "Blue"]	11	
["Blue", "Blue", "Blue", "Red", "Red", "Red"]	13	

Note:

- Only change coloring pencils if the next color is different to the color before it.
- Return a number in seconds.

Task 09(CP):

Your local bank has decided to upgrade its ATM machines by incorporating motion sensor technology. The machines now interpret a series of consecutive dance moves in place of a PIN number.

Create a program that converts a customer's PIN number to its dance equivalent. There is one dance move per digit in the PIN number. A list of dance moves is given below.

MOVES = ["Shimmy", "Shake", "Pirouette", "Slide", "Box Step", "Headspin", "Dosado", "Pop", "Lock", "Arabesque"];

Notes:

- Each dance move will be selected from a list by index based on the current digit's value plus that digit's index value. If this value is greater than the last index value of the dance list, it should cycle to the beginning of the list.
- Valid input will always be a string of four digits. Output will be on the console.
- If the input is not four valid numbers, output the string, "Invalid input."

Test Cases:

"0000" → "Shimmy", "Shake", "Pirouette", "Slide"

"3856" ➞ "Slide", "Arabesque", "Pop", "Arabesque"
"9999" ➞ "Arabesque", "Shimmy", "Shake", "Pirouette"
"32a1" ➞ "Invalid input."

Project Version 01:

Use parallel arrays to implement the CRUD (Create, Retrieve, Update and Delete) options for 1 user of your Project with menus.
For example: In the case of the UAMS System.

Menu for the Admin

1. Add Student (Create)
2. Search a Student by Name (Retrieve)
3. Change the Roll Number of the Student (Update)
4. Delete a Student by Name (Delete)

Good Luck and Best Wishes !!

Happy Coding ahead :)