Programming Fundamentals

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):

Write a C++ function named isLengthEven 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 true

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task1.exe
Enter a String: apples
1
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task1.exe
Enter a String: pears
0
```

Task 02(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".
Prototype of the function is as follows:

```
s  ontainsSeven(int numbers[], int size)
```

Programming Fundamentals

Programming Day - Week 09

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

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task2.exe
Enter the size of Array: 5
Enter Element 1: 2
Enter Element 2: 55
Enter Element 3: 97
Enter Element 4: 5
Enter Element 5: 3
Boom!

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task2.exe
Enter the size of Array: 3
Enter Element 1: 2
Enter Element 2: 4
Enter Element 3: 5
there is no 7 in the array
```

Task 03(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.
Test Cases:
areAllElementsIdentical(["@", "@", "@", "@"], 4) ➞ true
areAllElementsIdentical(["abc", "abc", "abc", "abc"], 4) ➞ true
areAllElementsIdentical(["SS", "SS", "SS", "SS"], 4) ➞ true
areAllElementsIdentical(["&&", "&", "&&&", "&&&&", "&"], 5) ➞ false
areAllElementsIdentical(["SS", "SS", "Ss"], 3) ➞ false

Programming Fundamentals

Programming Day - Week 09

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task3.exe
Enter the size of Array: 4
Enter Element 1: w
Enter Element 2: w
Enter Element 3: w
Enter Element 4: w
1
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task3.exe
Enter the size of Array: 3
Enter Element 1: sad
Enter Element 2: had
Enter Element 3: mad
0
```

Task 04(CP):

Write a C++ function 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.

Function prototype is:

```
void evenOddTransform(int arr[], int size, int n)
```

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]

[-20, -20, -20]

Enter number of times even-odd
transformation need to be done:
10

Enter the array: [1, 2, 3] [3, 0, 5]

Programming Fundamentals

Programming Day - Week 09

Enter number of times even-odd transformation need to be done: 1	

```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task4.exe
Enter the size of Array: 3
Enter Element 1: 3
Enter Element 2: 4
Enter Element 3: 9
Enter number of times even-odd transformation need to be done: 3
[9, -2, 15]

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task4.exe
Enter the size of Array: 3
Enter Element 1: 0
Enter Element 2: 0
Enter Element 3: 0
Enter number of times even-odd transformation need to be done: 10
[-20, -20, -20]

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task4.exe
Enter the size of Array: 3
Enter Element 1: 1
Enter Element 2: 2
Enter Element 3: 3
Enter number of times even-odd transformation need to be done: 1
[3, 0, 5]
```

Task 05(CP):

Write a C++ function that is Given two strings, find the number of common characters between them and then return 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".


```
G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task5.exe
Enter the first string: aabcc
Enter the second string: adcaa
Number of common characters: 3

G:\Semesters\Programming Fundamentals (Fall 2023)\Week 9\PD Tasks>Task5.exe
Enter the first string: hello
Enter the second string: hellohello
Number of common characters: 5
```

Task 06(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"]

Programming Fundamentals

Programming Day - Week 09

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

```
int coloringTime(string cols[], int size)
```

Task 07(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"];

Programming Fundamentals

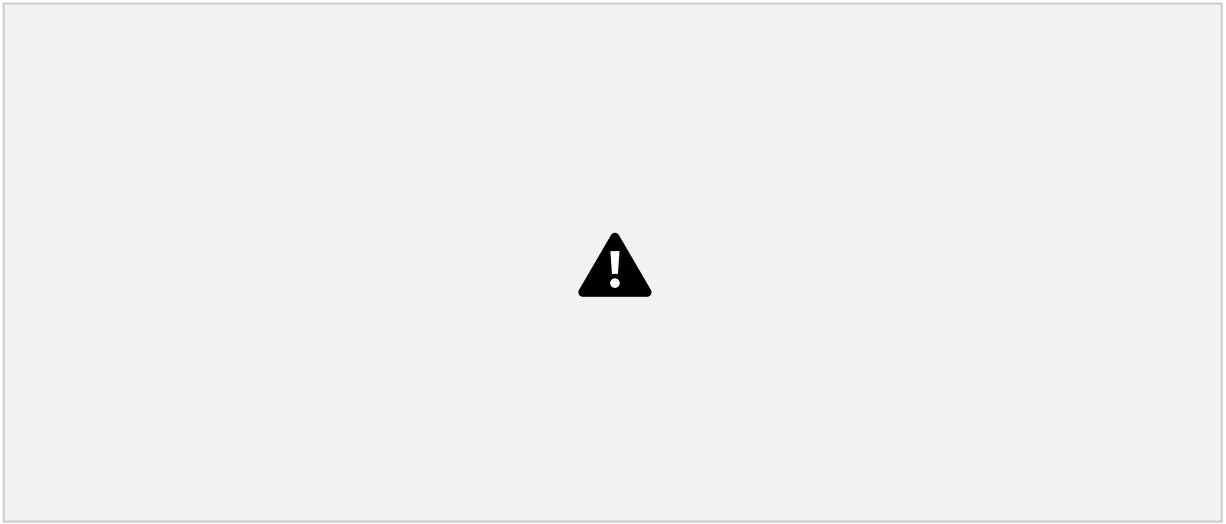
Programming Day - Week 09

- 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." ●

Function prototype should be

```
void convertPINToDance(string pin)
```

Test Cases:
"0000" ⟶ "Shimmy, Shake, Pirouette, Slide"
"3856" ⟶ "Slide, Arabesque, Pop, Arabesque"
"9999" ⟶ "Arabesque, Shimmy, Shake, Pirouette"
"32a1" ⟶ "Invalid input."



Task 08(CP):
Given what is supposed to be typed and what is actually typed, write a function that

returns the broken key(s). The function looks like:
findBrokenKeys(correct phrase, what you actually typed)

Examples

findBrokenKeys("happy birthday", "hawwy birthday") → "p"
findBrokenKeys("starry night", "starrq light") → "yn"
Keys("beethoven", "affthoif5") → "bevn"

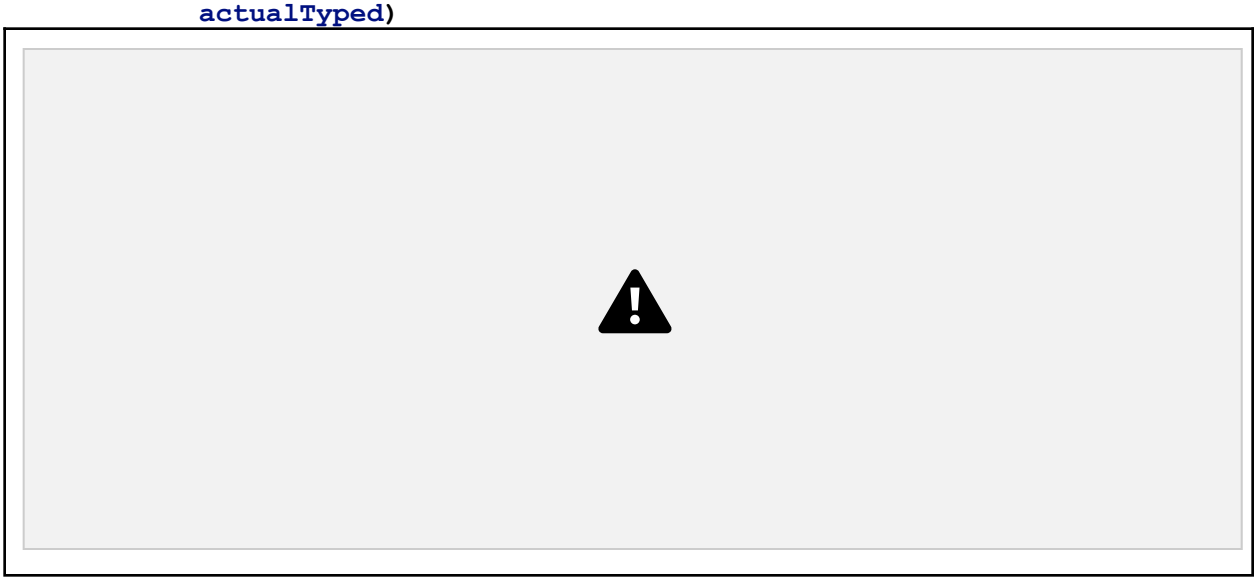
Programming Fundamentals

Programming Day - Week 09

Notes

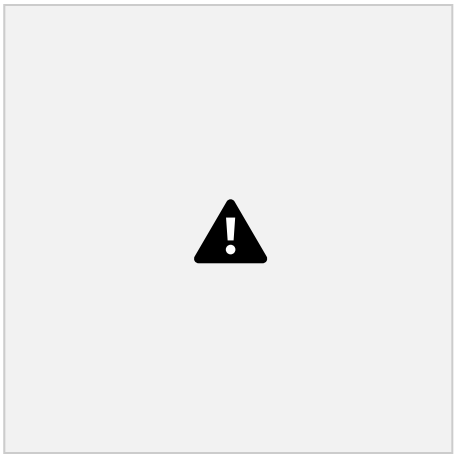
- Broken keys should be ordered by when they first appear in the sentence.
- Only one broken key per letter should be listed.
- Letters will all be in lower case.
- Function prototype should be

```
string findBrokenKeys(string correctPhrase, string
actualTyped)
```



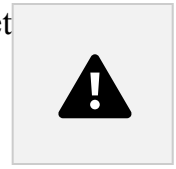
Task 09(CP):

Given an *array of words*, return the **longest word** which can fit on a *7 segment display*.



- Letters which do not fit on a *7 segment display* are **k**, **m**, **v**, **w** and **x**.

fore, do not count words which include these let



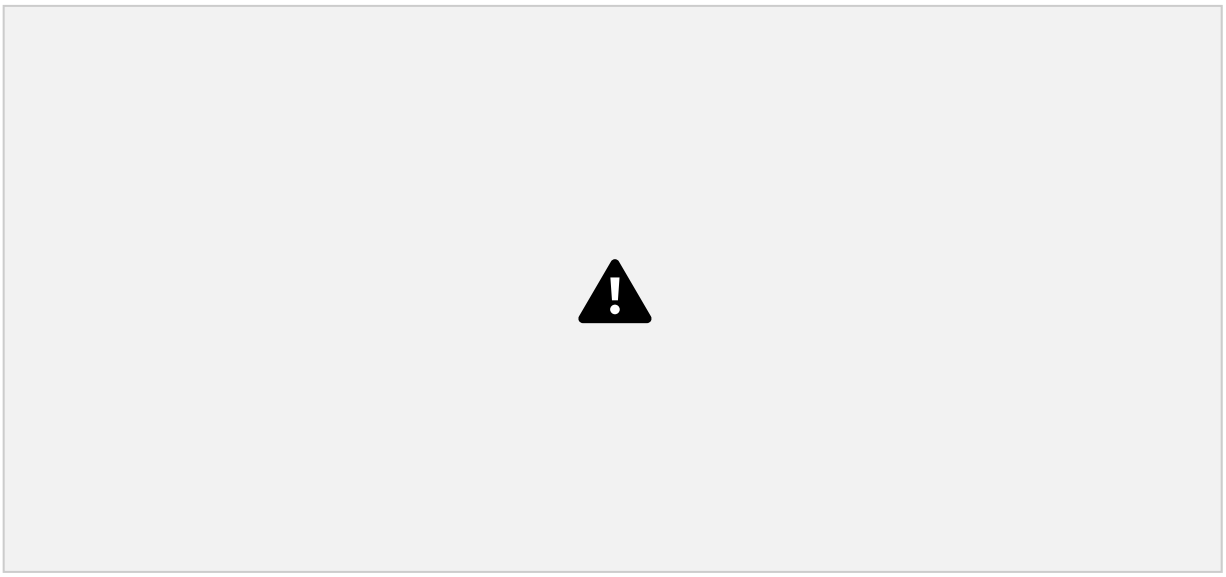
Programming Fundamentals

Programming Day - Week 09

Examples

longest7SegmentWord(["knighthood", "parental", "fridge", "clingfilm"], 4) → "parental"
longest7SegmentWord(["coding", "crackers", "celebration"], 3) → "celebration"
longest7SegmentWord(["velocity", "mackerel", "woven", "kingsmen"], 4) → "" **Notes**

- All words will be given in lowercase.
- Return an *empty string* if no words are eligible (see example #3). ● If multiple valid words have the same length, return the **first word that appears**.



Good Luck and Best Wishes !!

Happy Coding ahead :)