IT230 Lab 3
Score: _____/20 v1.1

In this lab, you'll use everything you've learned so far to analyze the efficiency of practical code samples that could be found in real-world codebases.

All portions of this lab will be written in C++, using the C++20 specification.

Specifications

Complete all functions by following the instructions below. Answer the questions in the Canvas "quiz".

Constraints

- Do not modify the main.cpp file.
- Do not modify the source.h file.
- Do not modify function prototypes/headers.

Tips

- If you see an error including "nan", that means "not a number". You typically see this if attempting to divide by 0, among other things.
- If you want to return a reference type, it must be allocated on the heap. Variables declared in the stack are automatically released when the function goes out of scope.

Submission

The submission for this lab is the commit ID you want to be graded. You will submit the commit ID via the Canvas "quiz". If you need help finding the commit ID, ask for help.

10 points will be assigned by correctness and your observable understanding as reflected in your code. The other 10 points will come from the Canvas "quiz".

Remember to write efficient code. Optimize your code wherever possible. Points will be deducted for unoptimized code.

Mean of Even Numbers

The `meanOfEvenNumbers()` method accepts an array of numbers and returns the mean (average) of all its even numbers.

Word Builder

The `wordBuilder()` function uses an algorithm that collects every combination of two-character strings built from an array of single characters. For example, given the array: ["a", "b", "c", "d"], you'd return a new array containing the following string combinations:

```
[
'ab', 'ac', 'ad', 'ba', 'bc', 'bd',
'ca', 'cb', 'cd', 'da', 'db', 'dc'
]
```

The following link may help you:

https://www.geeksforgeeks.org/how-to-convert-a-single-character-to-string-in-cpp/

For this problem, you may want to consider using a vector: https://www.geeksforgeeks.org/vector-in-cpp-stl/
(Note, you still need to return a string array.)

Keep in mind, repeat characters are not allowed. For example "aa" would not be allowed.

Word Builder – Three Character Strings

What would happen if you modified your algorithm to compute each combination of three-character strings? That is, for an example array of ["a", "b", "c", "d"], your function would return the array:

```
[
'abc', 'abd', 'acb',
'acd', 'adb', 'adc',
'bac', 'bad', 'bca',
'bcd', 'bda', 'bdc',
'cab', 'cad', 'cba',
'cbd', 'cda', 'cdb',
'dab', 'dac', 'dba',
'dbc', 'dca', 'dcb'
]
```

This is precisely what you'll do for the `wordBuilderTC()` function.

For this problem, you may want to consider using a vector. (Note, you still need to return a string array.)

Keep in mind, repeat characters are not allowed. For example "aaa" or "aab" would not be allowed.

Count the Ones

Here's an algorithm where the Big O is different from what it seems at first glance. The `countTheOnes()` function accepts an array of arrays (or rather, a vector or vectors), where the inner arrays contain 1's and 0's. The function then returns how many 1's there are.

So, for this example input:

```
[
[0, 1, 1, 1, 0],
[0, 1, 0, 1, 0, 1],
[1, 0]
]
```

your function will return 7, since there are seven 1's.