COP 3275C Assignment 7 -

Assignment purpose: This assignment will help you practice using strings and character arrays, performing operations on static and dynamic arrays, implementing basic search techniques.

1. **Strings:**
   - First name, reversed first name, uppercase first name, phrase, reversed phrase, uppercase phrase
2. **Dynamic String array**
   - NATO phonetic alphabet strings to match the name and phrase, see the sample output
3. **Dynamic integer array**
   - ASCII values to match the original name , uppercase name, original phrase, uppercase phrase

For this assignment you will practice arrays, character arrays, and strings. You will have the user enter their first name

- create a reversed version of the first name
- an uppercase version of the first name
- a string array of the NATO Phonetic alphabet that relates to the first name
- An ASCII value array that relates to the original first name and calculate the average
- An ASCII value array that relates to the uppercase first name and calculate the average

You will have the user enter a short phrase or title (example movie or book title)

Use the same functions as you did for the first name to do the following:

- create a reversed version of the phrase
- an uppercase version of the phrase
- a string array of the NATO Phonetic alphabet that relates to the phrase
- An ASCII value array that relates to the original phrase and calculate the average
- An ASCII value array that relates to the uppercase phrase and calculate the average

The user should be able to continue to enter phrases and you will create all the phrase versions as long as the user enters' y' to continue, see the template and sample output

- 3 WAYS to reverse a copy of a word

  1. Make a copy and then use : ***reverse(word.begin(),word.end());***
     //string functions begin and end
     ```
     //include this:
     #include <algorithm> //for the reverse function
     ```

  2. Make a copy and reverse the copy using a loop(loop halfway though the array):

     ```
     int len = word.length();
     for (int i = 0; i < len/2 ; i++)
     {
            char temp = word[i];
            word[i] = word[len - 1 - i];
            word[len - 1 - i] = temp;
     }
     ```

3. Assign the reverse to a separate array character by character

```
int len = word.length();
string reverse;
for (int i = 0; i < len; i++)
{
       reverse[i] = word[len-1-i];
}
```

- Phrase input with white space: declare ask and get a short phrase - use C++ getline (to have the user enter a phrase with white spaces.

```
string phrase;
cin.ignore(); //ignore the earlier enter
getline(cin, phrase);
```

Use programmer defined functions to break up your code and reuse features.

Suggested functions:

```
#include <algorithm> //reverse function
#include <iostream>
#include <string>
#include <cctype>

using namespace std;


//input: const reference string
//return the reverse of the string
string ReverseIt(const string& word);

//input: const reference string
//return the string in all capital letters
string MakeUpper(const string& word);

//input: const reference string and a string pointer
//dynamically create a sting array and fill it with the NATO phonetic strings
//print the string array
//delete the new string array
void StringArray(const string& word, string* Nato); //calls MakeNato and PrintStringArray

//input: character by value and string by reference
//assign the NATO phonetic string that matches the character (A through Z)
//for any other character assign space " "
void MakeNato(char letter, string& word);

//input: string array and the length of the array
//print the strings in the array in a column
void PrintStringArray(int len, string Nato[]);

//input: const reference string and a pointer to an integer
//dynamically create an integer array and fill it with the corresponding numerical (ASCII) value
//print the integer array
//delete the new integer array
void MakeASCII(const string& word, int* nums);//calls PrintNumberArray and CalculateAverage
```

```
//input: integer array and the length of the array
//print the integers in the array in a row separated by spaces
void PrintNumberArray(int len, int nums[]);

//input: integer array and the length of the array
//calculate and return the average of the numbers in the array
double CalculateAverage(int len, int nums[]);
```

Additional instructions:

- Be sure to comment your code
- Be sure to use function prototypes above main with function definitions below main
- Be sure to include comments on the function prototypes as well as the function definitions and throughout the code to make it easier to read and understand.
- Be sure to use descriptive variable names
- Be sure to use descriptive function names
- Submit ONE source code file: lastname_A7.cpp
- Include a program header COMMENT with the following information:
  - Name, date, course, and a brief description of the assignment
- Test your code in an IDE before submitting
- The file names must match the assignment
- The code must be submitted on time in order to receive credit (11:59 PM on the due date)
  - NOTE: there is a grace period until 8:00 AM the following day
- The assignment allows multiple submissions you may make revisions and resubmit until the final due date
- **Late submissions (after 8 AM or sent by email) will not be accepted or graded**

Modifying data and submitting it as your own is a fraudulent practice—specifically, plagiarism—and is no different than copying paragraphs of information from a book or journal article and calling it your own (make sure that you work independently and submit only your own work)

This programming assignment is individual work, sharing code is considered cheating,

The use of AI to assist in this assignment is prohibited.

NATO phonetic alphabet
- **A** - Alfa
- **B** - Bravo
- **C** - Charlie
- **D** - Delta
- **E** - Echo
- **F** - Foxtrot
- **G** - Golf
- **H** - Hotel
- **I** - India
- **J** - Juliet
- **K** - Kilo
- **L** - Lima
- **M** - Mike
- **N** - November
- **O** - Oscar
- **P** - Papa
- **Q** - Quebec
- **R** - Romeo
- **S** - Sierra
- **T** - Tango
- **U** - Uniform
- **V** - Victor
- **W** - Whiskey
- **X** - X-ray
- **Y** - Yankee
- **Z** – Zulu

```
Sample output (1):

Enter your first name: Tami

original name: Tami
reversed name: imaT
uppercase name: TAMI

NATO phonetic version:
       Tango
       Alpha
       Mike
       India

ASCII version: Tami
84 97 109 105
The average is 98.75


ASCII version: TAMI
84 65 77 73
The average is 74.75



Enter a short phrase or title: Pride and Prejudice

original phrase: Pride and Prejudice
reversed phrase: ecidujerP dna edirP
uppercase phrase: PRIDE AND PREJUDICE

NATO phonetic version:
       Papa
       Romeo
       India
       Delta
       Echo

       Alpha
       November
       Delta

       Papa
       Romeo
       Echo
       Juliet
       Uniform
       Delta
       India
       Charlie
       Echo

ASCII version: Pride and Prejudice
80 114 105 100 101 32 97 110 100 32 80 114 101 106 117 100 105 99 101
The average is 94.4211


ASCII version: PRIDE AND PREJUDICE
80 82 73 68 69 32 65 78 68 32 80 82 69 74 85 68 73 67 69
The average is 69.1579


would you like to enter another phrase (y or n)? y


Enter a short phrase or title: Harry Potter and the Sorcerer's Stone

original phrase: Harry Potter and the Sorcerer's Stone
reversed phrase: enotS s'rerecroS eht dna rettoP yrraH
uppercase phrase: HARRY POTTER AND THE SORCERER'S STONE

NATO phonetic version:
       Hotel
       Alpha
       Romeo
       Romeo
       Yankee

       Papa
       Oscar
       Tango
       Tango
```

```
        Echo
        Romeo

        Alpha
        November
        Delta

        Tango
        Hotel
        Echo

        Sierra
        Oscar
        Romeo
        Charlie
        Echo
        Romeo
        Echo
        Romeo

        Sierra

        Sierra
        Tango
        Oscar
        November
        Echo

ASCII version: Harry Potter and the Sorcerer's Stone
72 97 114 114 121 32 80 111 116 116 101 114 32 97 110 100 32 116 104 101 32 83 111 114 99 101 114
101 114 39 115 32 83 116 111 110 101
The average is 93.4054


ASCII version: HARRY POTTER AND THE SORCERER'S STONE
72 65 82 82 89 32 80 79 84 84 69 82 32 65 78 68 32 84 72 69 32 83 79 82 67 69 82 69 82 39 83 32 83
84 79 78 69
The average is 70.0541


would you like to enter another phrase (y or n)? n
```

```
SAMPLE OUTPUT (2)

Enter your first name: Daniela

original name: Daniela
reversed name: aleinaD
uppercase name: DANIELA

NATO phonetic version:
        Delta
        Alpha
        November
        India
        Echo
        Lima
        Alpha

ASCII version: Daniela
68 97 110 105 101 108 97
The average is 98


ASCII version: DANIELA
68 65 78 73 69 76 65
The average is 70.5714
```

```
Enter a short phrase or title: spongebob squarepants

original phrase: spongebob squarepants
reversed phrase: stnaperauqs bobegnops
uppercase phrase: SPONGEBOB SQUAREPANTS

NATO phonetic version:
       Sierra
       Papa
       Oscar
       November
       Golf
       Echo
       Bravo
       Oscar
       Bravo

       Sierra
       Quebec
       Uniform
       Alpha
       Romeo
       Echo
       Papa
       Alpha
       November
       Tango
       Sierra

ASCII version: spongebob squarepants
115 112 111 110 103 101 98 111 98 32 115 113 117 97 114 101 112 97 110 116 115
The average is 104.667


ASCII version: SPONGEBOB SQUAREPANTS
83 80 79 78 71 69 66 79 66 32 83 81 85 65 82 69 80 65 78 84 83
The average is 74.1905


would you like to enter another phrase (y or n)? y


Enter a short phrase or title: happy fourth of july

original phrase: happy fourth of july
reversed phrase: yluj fo htruof yppah
uppercase phrase: HAPPY FOURTH OF JULY

NATO phonetic version:
       Hotel
       Alpha
       Papa
       Papa
       Yankee

       Foxtrot
       Oscar
       Uniform
       Romeo
       Tango
       Hotel

       Oscar
       Foxtrot

       Juliet
       Uniform
       Lima
       Yankee
```

```
ASCII version: happy fourth of july
104 97 112 112 121 32 102 111 117 114 116 104 32 111 102 32 106 117 108 121
The average is 98.55


ASCII version: HAPPY FOURTH OF JULY
72 65 80 80 89 32 70 79 85 82 84 72 32 79 70 32 74 85 76 89
The average is 71.35


would you like to enter another phrase (y or n)? n
```