Module 10:
Strings

COP2274

In-class Assignments

UF UNIVERSITY of FLORIDA

M10A C-string

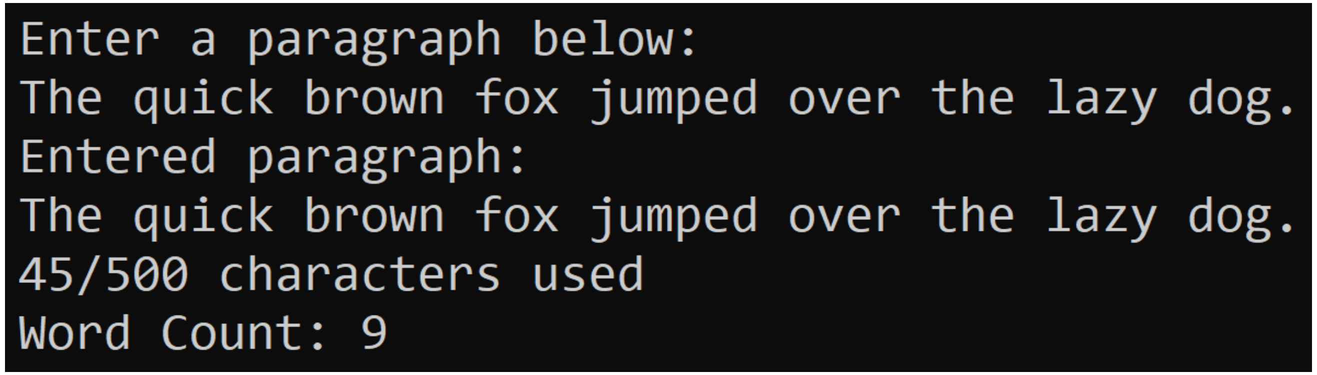
1. Create **a C-string** with a capacity of 501 characters (500 characters + 1 for the null character). The current maximum length of a paragraph is 500 characters.
2. Use the *getline() function* to read a paragraph from the user. Then, output the user-inputted paragraph, the character used, and how many words are written in the paragraph as shown in the test case.

Notes:

- *You may want to #include <cstring> for access to the strlen() function.*
- *You can assume that the user will always start with a letter not a space.*

M10A C-string

Test case



```
Enter a paragraph below:
The quick brown fox jumped over the lazy dog.
Entered paragraph:
The quick brown fox jumped over the lazy dog.
45/500 characters used
Word Count: 9
```

M10B Strings
(C-strings vs. string objects)

Write a program that utilizes **the standard class string** in conjunction with **C-strings**.

1. Prompt the user for a first and last name, storing into **each C-string**, respectively.
2. Prompt the user for a second first and last name, storing into **each string object**, respectively.
3. Make the first letter of each first and last name uppercase as shown in the test case.
4. Format the first name of the C-string and the string object to include only the first letter.
5. Concatenate the first and last names of the C-string, and concatenate the first and last names of the string.

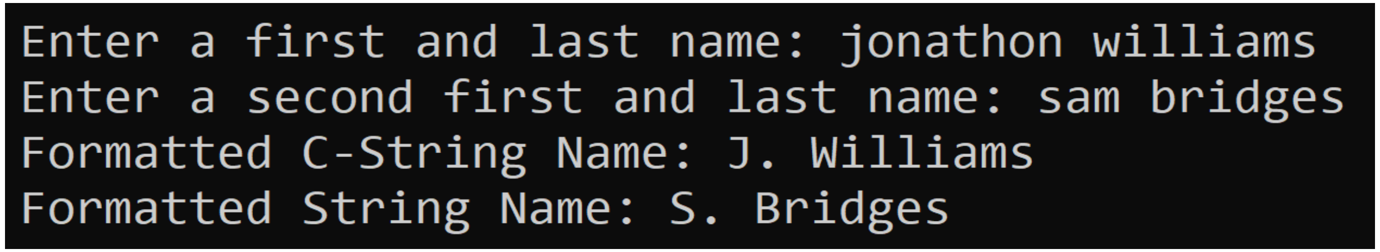
M10B Strings (C-strings vs. string objects)

6. Display results as shown in the test case.

Notes:

- *You can create the two C-strings with a capacity of 20.*
- *You may want to #include <cctype> for access to the toupper() function to convert a single character to uppercase.*

Test case



```
Enter a first and last name: jonathon williams
Enter a second first and last name: sam bridges
Formatted C-String Name: J. Williams
Formatted String Name: S. Bridges
```

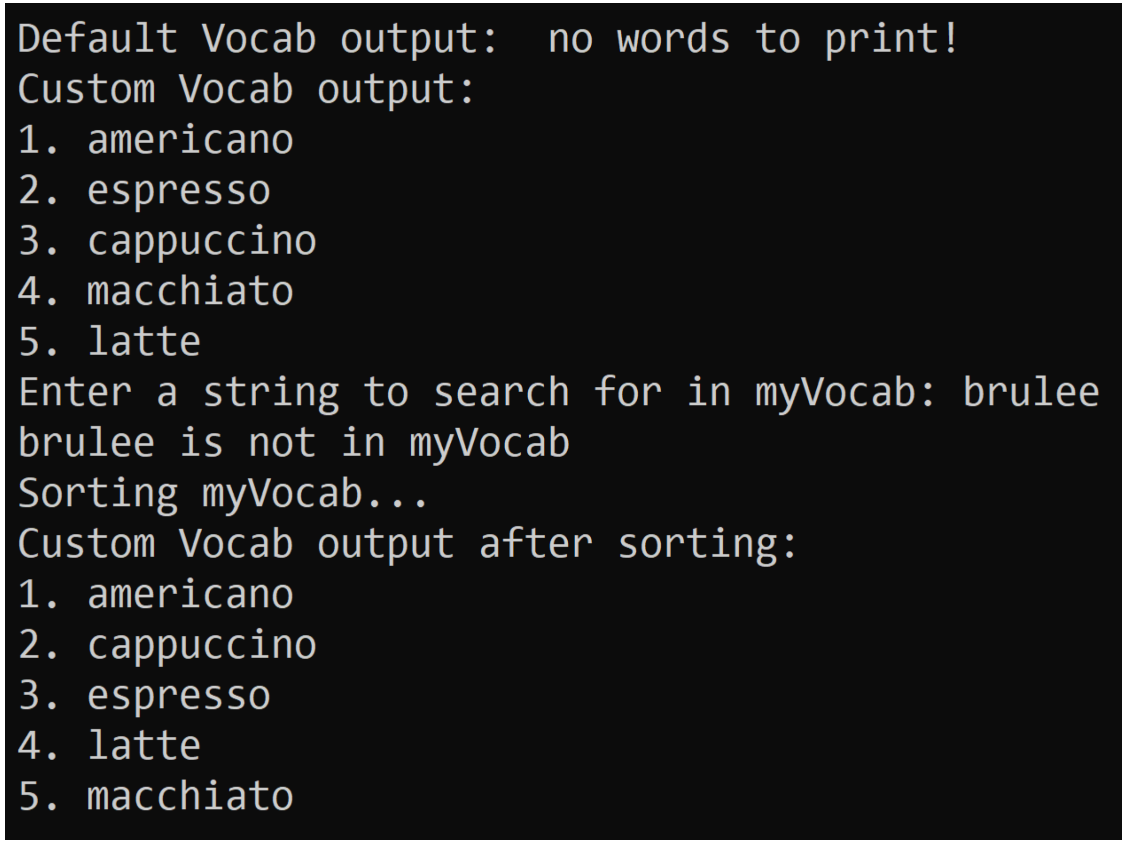
M10C Class called Vocabulary (with an array of string objects)

Write a **class called Vocabulary** that stores **an array of string objects**.

1. Write default and custom constructor. The maximum number of words that can be stored is 20.
2. Write a member function that prints the words in the Vocabulary.
3. Write a member function that checks if a word is in the Vocabulary. (returns true or false)
4. Write a member function that sorts the words in the Vocabulary lexicographically (alphabetically).
5. In main: Test Vocabulary object by initializing it with a bunch of words (strings). Also make a default Vocabulary.
6. Print the Vocabulary objects and prompt the user for a string. Tell the user if the string is in the vocabulary.
7. Sort the Vocabulary. Print the Vocabulary after sorting.

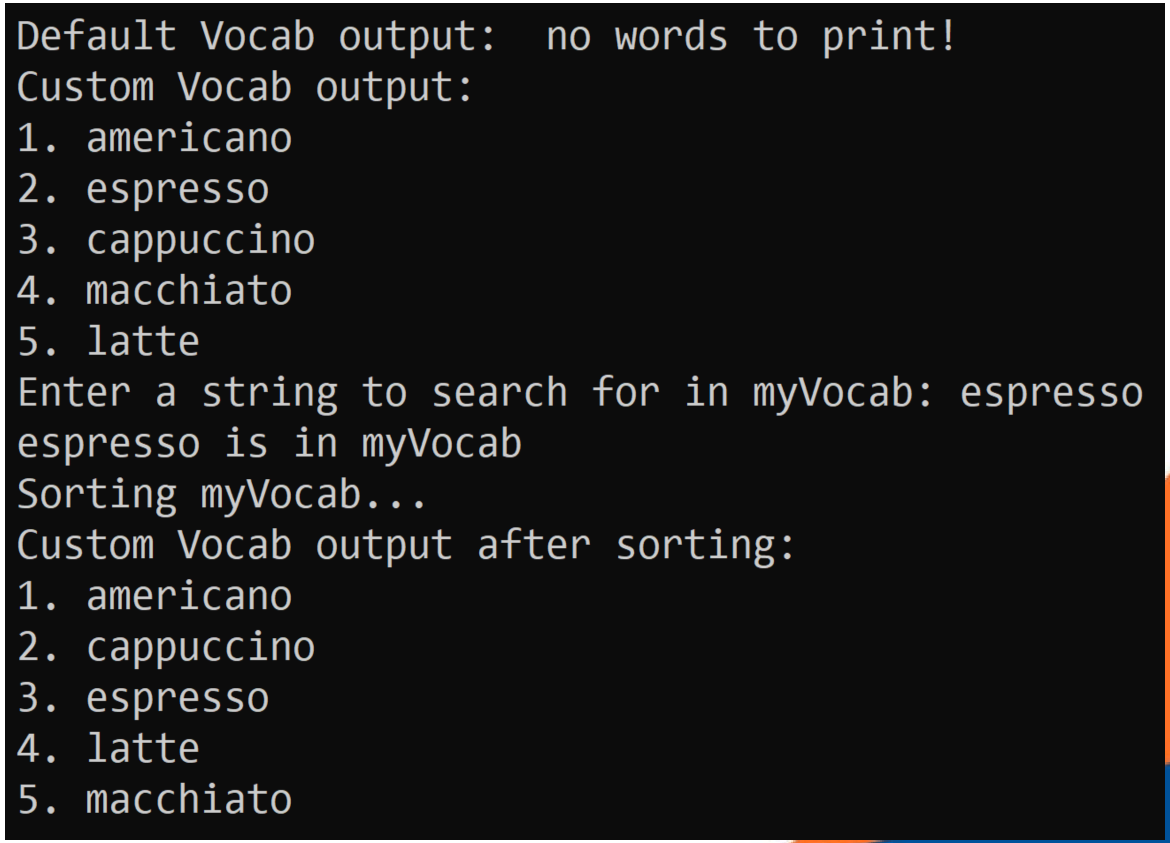
M10C Class called Vocabulary (with an array of string objects)

Test case 1



```
Default Vocab output:  no words to print!
Custom Vocab output:
1. americano
2. espresso
3. cappuccino
4. macchiato
5. latte
Enter a string to search for in myVocab: brulee
brulee is not in myVocab
Sorting myVocab...
Custom Vocab output after sorting:
1. americano
2. cappuccino
3. espresso
4. latte
5. macchiato
```

Test case 2



```
Default Vocab output:  no words to print!
Custom Vocab output:
1. americano
2. espresso
3. cappuccino
4. macchiato
5. latte
Enter a string to search for in myVocab: espresso
espresso is in myVocab
Sorting myVocab...
Custom Vocab output after sorting:
1. americano
2. cappuccino
3. espresso
4. latte
5. macchiato
```