

1-Asks the user to enter any sequence of characters  
2-The program should display all the lower-case vowels that are present and the number of times each vowel occurs  
3-The program should display all the upper-case vowels that are present and the number of times each vowel occurs  
4-The program should display the upper-case vowel which appears most frequently in the user input AND the number of times it appeared.  
5-The program should display the lower-case vowel which appears most frequently in the user input AND the number of times it appeared.  
6-The user must be asked if he/she wants to continue entering values or quit.  
7-You must demonstrate effective use of the new and delete operators.

THIS IS THE POINT OF THE QUIZ

Remember to comment your code

8-Use meaningful or mnemonic variable names

9You must use a pointer to a variable of type char to store the user input, and you MUST allocate the EXACT amount of memory to store all the characters entered by the user. For example:

If the user types Awq, then you need to use the new operator to allocate dynamic memory for a pointer to a char of size 3: UserInputCharArray = new char[3];

If the user types 300 letters, then you need to use the new operator to allocate memory for a pointer to a char of size 300: UserInputCharArray = new char[300];

In order to accomplish this, you must capture the user input one char at a time using a loop. As you are capturing the chars you need to be clever enough to count the chars as you are removing them from the input stream AND somehow use new and delete operators to create And destroy pointers (plural) to store the chars as you are removing them from the input stream. Remember that the new and delete operators work at RUN TIME NOT COMPILE time! So use them to allocate JUST THE EXACT AMOUNT OF MEMORY FOR THE ENTRY.

Below is an example of

WHAT NOT TO DO

!!!

```
const in ARRAY_SIZE =100;//bad idea
char *UserInputCharArray = nullptr;//this is correct
//allocate memory to the pointer
UserInputCharArray = new char[ARRAY_SIZE]; //you are allocating at
compile time the size of the array; another VERY BAD idea!
```

This code above is NOT allowed because you are assuming the user will enter a string of exactly 100 chars. Unless you have a crystal ball and can guess that every user will enter a string of 100 chars every single time. Others reasons why this is not allowed are:

If the user enters a four char entry like "gaga", then you are wasting memory, by allocating memory for an array of 100 chars

If the user enters a four hundred char entry by typing "gaga" 100 times; then you are notallocating enough memory, to capture the entire entry!

HINT

```
char aChar;//declare a variable of type char
// Get a string. Prompt user
cout << "Enter a string: ";
//capture userInput, one char at a time
aChar = cin.get();
while (aChar != 10)
{
//your code goes here
}
```

GENERAL RESTRICTIONS

No global variables

No labels or go-to statements

No infinite loops, examples include:for(;;)

while(1)

while(true)

do{//code}while(1);

No break statements to exit loops

SPECIFIC RESTRICTIONS

You may use string literals, but do NOT use variables of type string;

this will cost you the quiz!