Homework 5: Strings

CS16 - Winter 2021

Due: Thursday, February 11, 2021 (11:59 PM PST)
Points: 100
Name: ____
Homework buddy: ____

- You may collaborate on this homework with **at most** one person, an optional "homework buddy."
- **Submission instructions:** All questions are to be written (either by hand or typed) *in the provided spaces* and turned in as a single PDF on Gradescope. If you submit handwritten solutions write legibly. We reserve the right to give 0 points to answers we cannot read. When you submit your answer on Gradescope, **be sure to select which portions of your answer correspond to which problem** and clearly mark on the page itself which problem you are answering. We reserve the right to give 0 points to submissions that fail to do this.

1. (10 points) Which of these are correct usage (syntax) of a single statement on a string variable called `message`, and which of these are incorrect usage (and *very briefly* **why**). Variables `n` and `m` are `int` types.

a. (2 points) `message.erase(n, m);`

b. (2 points) `message = message.erase(n, m);`

c. (2 points) `cout << message.find(n);`

d. (2 points) `message.size() = n;`

e. (2 points) `cout << message.rfind("x");`

2. (10 points) The following code takes in a string input from the user and performs an integer multiplication, as seen in the example run here. Note that the input string will contain the asterisk character `'*'`:

```
Enter 2 integer numbers to be multiplied, like this: num1*num2: 15*3
The answer is: 45
```

Complete the missing code below that performs this task (it can be done in 2 lines, but you can use more if you like).

```
string s; int k(0);
cout << "Enter 2 integer numbers to be multiplied, like this: num1*num2: ";
cin >> s;




cout << "The answer is: " << k << endl;
```

3. (20 points) Given the declaration of a C-string variable, where `MAX` is a defined constant: `char buffer[MAX];`

The C-string variable `buffer` has previously been assigned in code not shown here. For correct C- string variables, the following loop reassigns all positions of `buffer` the value ‘z’, leaving the length the same as before. Assume this code fragment is embedded in an otherwise complete and correct program. Answer the questions following this code fragment:

```
int index = 0;
while (buffer[index] != '\0') {
   buffer[index] = 'z';
   index++;
}
```

a. (10 points) Explain how this code can destroy memory beyond the end of the array.

b. (10 points) Modify this loop to protect against inadvertently changing memory beyond the end of the array.

4. (20 points) Show the output produced when the following code (entire program not shown) executes. *If there is an error in this code*, point it out and explain why it is not correct. You are encouraged to also try to compile this to verify your results.

```
string name = "Porcupine Tree";
cout << "NAME = " + name << endl;
cout << name.length() << endl;
name.erase(8, 6);
cout << name << endl;
name.append("Dean WD Morgan");
cout << name << endl;
name.insert(22, "@TWD");
cout << name << endl;
name.replace(23, 3, "The WD");
cout << name << endl;
cout << name.find("WD") << endl;
cout << name.rfind("WD") << endl;
cout << name.rfind("cupi") << endl;
for (int i = name.length(); i > 20; i--) {
   cout << name[i-1];
   cout << endl;
}
```

5. (20 points) Write the full definition of a function called `FunString()` that takes a string argument and does 2 things: (1) it prints the *second* half of the string backwards (while still printing the first half normally), and (2) it reports on how many words the original string has (assume a word is separated with space characters). For example, if the argument is “All the strings”, the function should print out “All thesgnirts” on one line and then the number `3` on the next line.

6. (20 points) Write a full definition for a function called `IsLoud()` that takes in a string argument and checks if each character in the string is an uppercase character *or* a `'!'`. If all characters pass this test, then the function returns true, otherwise it returns false.