CS 31 Worksheet Week 4

This worksheet is entirely optional, and meant for extra practice. Some problems will be more challenging than others and are designed to have you apply your knowledge beyond the examples presented in lecture, discussion or projects. All exams will be done on paper, so it is in your best interest to practice these problems by hand and not rely on a compiler. Solutions are written in red. The solutions for programming problems are not absolute, it is okay if your code looks different; this is just one way to solve the specific problem.

Concepts: While Loops, Do While Loops, String Traversal, Functions (Only pass-by-value)

Reading Problems

1) What does the following code snippet output, AND what does each one return?

```cpp
int mystery(int a, int b) {
    int count = 0;
    while (count < 2) {
        a = a + b/2;
        b = a + 5;
        cout << "a: " << a << " b: " << b << endl;
        count++;
    }
    return count;
}

int main() {
    int a = 5, b = 10;
    cout << "a: " << a << " b: " << b << endl;
    int c = mystery(a, b);
    cout << "a: " << a << " b: " << b << endl;

    if (a % c == 0) {
        return 1;
    }
    return 0;
}
```
2) What does the following code snippet output?

```cpp
#include <iostream>
using namespace std;

void mystery (char code, int j) {
    if (j % 3 == 0)
        cout << "I found " << code << " at index " << j;
    else
        cout << "meh too lazy rn" << endl;
    if (j % 1 == 0)
        cout << "!" << endl;
}

int main() {
    string message = "2319!:(";
    for (int i = 0; i != message.length(); i++) {
        mystery(message[i], i);
    }
}
```

Time: <10 min
Programming Problems

Note: Now that you have had some more in-class practice with functions and strings, it might be a good idea to revisit the problems from last week if you haven't figured them out already! :)

1) Write a function called reverse(int n) that reverses the digits in the parameter and returns its result. For example, if n is 927, it should return 729. Then, in a main() function, take a number from user input, and call the function reverse using that number you just received.

Example test cases:

Enter number: 123
321

Enter number: 001
1

Enter number: 0
0

Time: ~20 min
#include <iostream>

using namespace std;

int reverse(int n){
    // What we want to do is try to build up result digit by digit, reversing the process of Problem 1 on last week's worksheet!
    int result = 0;
    while (n >= 10){
        result += (n % 10);
        n /= 10;
    }
    return result;
}

int main(){
    int n;
    cout << "Enter number: " << endl;
    cin >> n;
    cout << reverse(n) << endl;
    return 0;
}
```cpp
result = result * 10;
    n = n / 10;
}
result += n;
return result;
}

int main(){
    int n = 0;
    cout << "Enter number: ";
    cin >> n;
    cout << reverse(n) << endl;
}

2) Write a function max3 which uses 3 parameters and returns the value of the largest.
   Test the function in a program that determines the largest of 3 quiz scores. If the
   score is more than 100, print a congratulatory message!

   Example test cases:

   Enter student 1's score: 105
   Enter student 2's score: 44
   Enter student 3's score: 56

   The highest score out of these was 105! Congratulations on your hard work!

   Time: ~10 min
   #include <iostream>

   using namespace std;
   int max3(int score1, int score2, int score3){
       // we use the fact that max(score1, score2, score3) = max( max(score1, score2), score3)

       int max;

       if (score1 >= score2){
           max = score1;
       } else{
```
max = score2;
}

if (score3 >= max){
    max = score3;
}

return max;
}

int main(){
    int one, two, three;
    cout << "Enter student 1's score: ";
    cin >> one;

    cout << "Enter student 2's score: ";
    cin >> two;

    cout << "Enter student 3's score: ";
    cin >> three;

    int max = max3(one, two, three);
    cout << "The highest score out of these was " << max << "! ";

    if (max > 100){
        cout << "Congratulations on your hard work!";  
    }

    cout << endl;
}

3. Given a string, write a function that can reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order. You can make it any kind of return value and have whatever parameters you need. However, explain your reasoning for your choices by using comments (remember, they are denoted with //)

Example:

Input: "Let's take LeetCode contest"

Output: "s'teL ekat edoCteeL tsetnoc"

Note: In the string, each word is separated by single space and there will not be any extra space in the string.
Reverse String

#include <iostream>
using namespace std;

string wordReverse(string str) {
    int length = 0; // keep track of the length of each word
    for (int i = 0; i <= str.size(); i++){  // we do <= to make sure we're still reversing the last word
        if (str[i] == ' ' || i == str.size()){
            // reverse the last 'length' characters
            // note that this begins with i - length and ends with i - 1
            for (int j = 0; j < length / 2; j++){
                // swap 1st and last, then 2nd and 2nd to last, etc.
                char temp = str[i - length + j]; // this is the j-th character in our word
                str[i - length + j] = str[i - j - 1]; // this is the j-th to last character in our word
                str[i - j - 1] = temp;
            }
            length = 0;
        } else{
            length++;
        }
    }
    return str;
}

int main() {
    string input;
    getline(cin, input);

    string result = wordReverse(input);

    cout << result << endl;
}

Alternate solution:

#include <iostream>
using namespace std;

string reverseSentence(string s) {
    string word = "";

    // Example usage:
    string sentence = "Hello, world!"
    string reversed = reverseSentence(sentence)
    cout << reversed << endl;
}
string sentence = "";
for(int i = 0; i < s.size(); i++){
    if(s[i]==' '){
        sentence += word + " ";
        word = "";
    }
    else{
        word = s[i] + word;
    }
}
sentence += word;
return sentence;
}
int main(){
    string input;
    getline(cin, input);
    string result = reverseSentence(input);
    cout << "" << result << "" << endl;
}

4. Implement a function named ToLowerCase that has a string parameter str, and returns the same string in lowercase.

Example 1:
Input: "Hello"
Output: "hello"

Example 2:
Input: "here"
Output: "here"

Example 3:
Input: "LOVELY Day!"
Output: "lovely day!"

Time: ~10 min

ToLowerCase

#include <iostream>

using namespace std;
string ToLowerCase(string str){
    for (int i = 0; i != str.size(); i++){
        str[i] = tolower(str[i]);
    }
    return str;
}
int main(){
    string input;
    getline(cin, input);

    string result = ToLowerCase(input);

    cout << result << endl;
}