Understanding:

The following is being asked of me:

- Create a string.cpp program that does the following:
  - Get's user input
  - prints the string one character at a time forward
  - prints the string one character at a time backward
  - counts how many letters there are in the string
  - discuss questions as listed
- Create a name.cpp program that does the following:
  - Get's user input (name in format First Middle Last)
  - Print name in Last, First Middle or Last, First M. or Last, First
- randNum.cpp
  - Print 10 random numbers
- loop.cpp
  - Compare two strings using the individual characters in each
  - discuss
- randNum2.cpp
  - Get two ints and print a random number between them. Do validation to make sure we are actually getting ints.
- wordGuessDesign.pdf
  - Design a project with explanations and pseudocode with given specifications.
- Upload the above to TEACH with this reflection

Design:

strings.cpp

headers and includes

using statement

```
int GetNumAlphaCharsInString(string my_str)
{
        //This function uses a for loop
        //to count the characters that are alpha

        for(…i<my_str.length()…)
        {
                if(isalpha(my_str.at(i)) { count++ }
        }

        return count;

}
```

```
void PrintStringBackward(string my_str)
{
	//This function uses a for loop
	//to count the characters that are alpha

	for(…i<my_str.length()…)
	{
		cout << my_str.at(my_str.length() - i - 1);
	}

}

void PrintStringForward(string my_str)
{
	//This function uses a for loop
	//to count the characters that are alpha

	for(…i<my_str.length()…)
	{
		cout << my_str.at(i);
	}

}

void main()
{
	//boilerplate code to get input not displayed
	//use getline

	PrintStringForward(my_string)
	PrintStringBackward(my_string)
	cout << GetNumAlphaCharsInString(my_string)


}
```

name.cpp

```
headers and includes

using statement

void PrintReformattedName(char name[])
{
   //Setting the splits to -1 allows us to see if they actually get set
   int split1 = -1;
   int split2 = -1;
   int len = strlen(name);
```

```
    //init cstrings to hold FLM names (not shown)

    for(int i = 0; i < STRING_LENGTH; i++)
    {
        if(name[i] == ' ' && split1 == -1)
        {
            split1 = i;
            continue;
        }
        if(name[i] == ' ' && split2 == -1)
        {
            split2 = i;
            continue;
        }
    }

   //if then statement detects if splits are set and knows if we have two or three names

    if(split1 != -1 && split2 != -1)
    {
        for(int i = 0; i < split1; i++)
        {

            int tmpLen = strlen(firstName);
            firstName[tmpLen] = name[i];
            firstName[tmpLen + 1] = '\0';

        }

        for(int i = split1 + 1; i < split2; i++)
        {
            //same as above
        }

        for(int i = split2 + 1; i < len; i++)
        {
            //same as above
        }

        cout << lastName << ", " << firstName << " " << middleName << endl;
    }
    else
    {
        //same deal, we will just skip split2
    }
}

int main()
```

```
{
   //Boilerplate code to get name not shown

   PrintReformattedName(name);

   return 0;
}
```

randNum.cpp

```
headers and includes

using statment

int main()
{
          for(loop to 10) { cout << rand() << endl; }

}
```

loop.cpp

```
headers and includes

using statement

bool StrComp(string string1, string string2)
{
   if(string1.length() != string2.length())
   {
      return false;
   }
   else
   {
      for(int i = 0; i < string1.length(); i++)
      {
         if(string1.at(i) == string2.at(i))
         {
            continue;
         }
         else
         {
            return false;
         }
      }

      return true;
```

```
    }

}

int main()
{
    //boilerplate create string and get user input

    bool same = StrComp(string1, string2);

    if(same)
    {
        cout << "The strings are the same." << endl;
    }
    else
    {
        cout << "The strings are different." << endl;
    }
    return 0;
}
```

randNum2.cpp

```
int main()
{
    srand(time(NULL));

    int lowerBound;
    int upperBound;

    // get input and do validation (not shown)

    int randomNumber = rand() % ((upperBound - lowerBound) + 1) + lowerBound;
    cout << randomNumber;


    return 0;
}
```

Testing

strings.cpp

Input	Expected output	Output
Hello	h e l l o o l l e h 5	Same as expected

name.cpp

Input	Expected output	Output
Nathan James Sosnovske	Sosnovske, Nathan James	Same as expected
Nathan Sosnovske	Sosnovske, Nathan	Same as expected
Nathan J Sosnovske	Sosnovske, Nathan J.	Same as expected

randnum.cpp

Input	Expected output	Output
N/A	Random numbers	Same as expected

loop.cpp

Input	Expected output	Output
hello hello	“Strings same”	Same as expected
hello goodbye	"Strings different”	Same as expected

randNum2

Input	Expected output	Output
1 5	Random number between 1 and 5 (inclusive)	Same as expected

Reflection

1. What did you learn? What problems did you encounter and how did you solve them?
    1. **I was able to find a pretty cool input validation routine on stack overflow for numGuess2. It could have been solved with things that I already knew, but it would have required a lot of boilerplate code. I also played around with pointers and pass by reference, which was fun.**
2. Was your understanding complete at the start of the project or did you learn something about the problem as you went?
    1. **I could have solved each problem and the project without any extra reading or research. I chose to do so in order to find more elegant solutions as discussed above.**
3. Was your initial design adequate for meeting all the requirements for the assignment or did you have to add to or alter your initial design to get it to meet all the requirements for the assignment?
    1. **My initial designs were satisfactory and simply required small tweaks in order to compile. They met all requirements.**
4. Did all your tests work out the way you expected, did you have to alter your design because of some failed tests, did you have to alter your tests because of some implementation details?
    1. **My tests all worked as expected.**
5. Did implementation go without any problems, were there details that were difficult to get working the way you wanted them to?

    1. **Pointers were kinda tricky, but that was a self-imposed problem. Otherwise, implementation from pseudo-code (which I basically write as real code without an ide) went smoothly.**
6. What techniques have we covered this week and in past weeks that helped you approach the problem, did you need to use outside sources to help solve the problem (list sites, books, or other materials that were helpful)? Does this project seem related to previous projects and do you see any names for future projects that it might be related to?
    1. **I used stack overflow for one question that I had. I also used the c++ library reference to learn how to use rand. I haven't had a need to do the reading or watch the lectures yet, so I'm not sure what in class material would have been useful.**