Assignment 5(Longest Common SubString):

Cpp code

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

int compute(string a, string b, int n, int m)
{

    if (m == 0 || n == 0)
    {
        return 0;
    }

    if (n == 1 && m == 1)
    {
        if (a[0] == b[0])
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }

    int dp[m + 1][n + 1];
    int final = 0;

    for (int i = 0; i < n + 1; i++)
    {
        for (int j = 0; j < m + 1; j++)
        {
            if (a[i - 1] == b[j - 1])
            {
                dp[i][j] = 1 + dp[i - 1][j - 1];
            }
            else
            {
                dp[i][j] = 0;
            }
            final = max(final, dp[i][j]);
        }
```

```
    }
    return final;
}
int main()
{
    string a = "ababc";
    string b = "babac";
    int n = a.length();
    int m = b.length();
    cout << compute(a, b, n, m);
    return 0;
}
```

- Test cases passed
- Completed on 9/3/23

Q/A:

1. How long did you spend on this assignment?
    a. 1day
2. Based on your effort, what letter grade would you say you earned?
    a. On a scale of 1 to 10. I would grade this as 10/10.
3. Based on your solution, what letter grade would you say you earned?
    a. On a scale of 1 to 10. I would grade this as 9/10.
4. Provide a summary of what doesn't work in your solution, along with an explanation of how you attempted to solve the problem and where you feel you struggled?
    a. The length of a common string grows linearly by visiting all states in the matrix as explained , we can use previous information diagonally by adding the values to the current value.