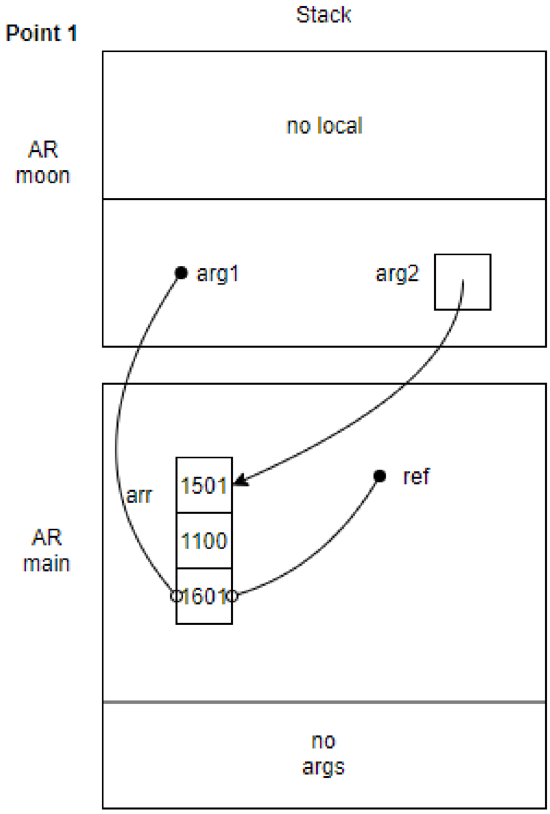
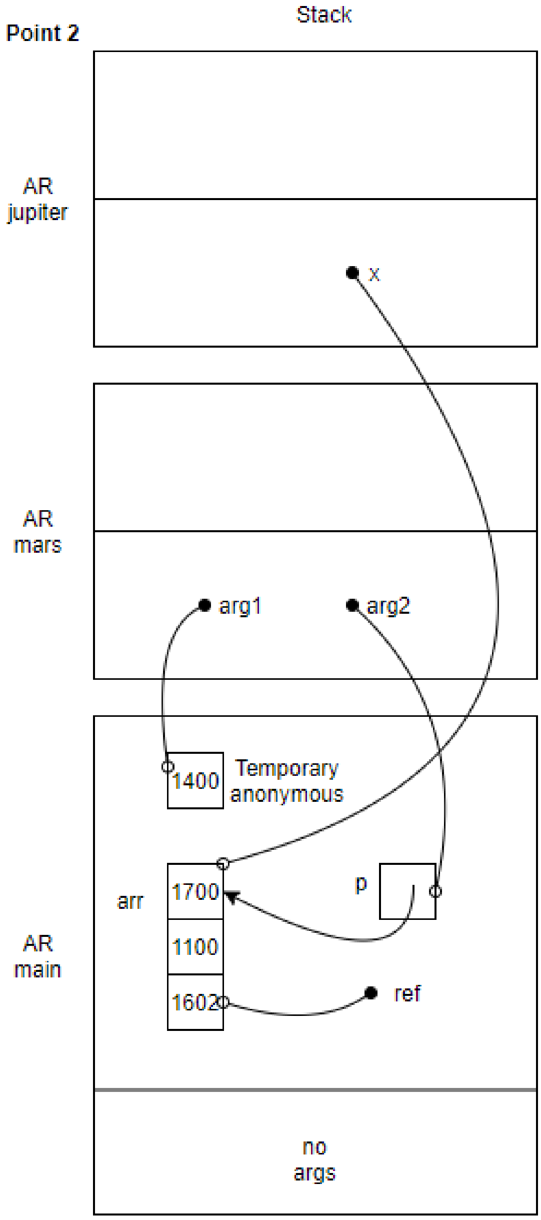
Lab 3 Report

Course: ENSF 619 - Fall 2020
Lab #: Lab 3
Student Name: Davis Allan, 10016543
Submission Date: Oct 9 2020

Exercise A

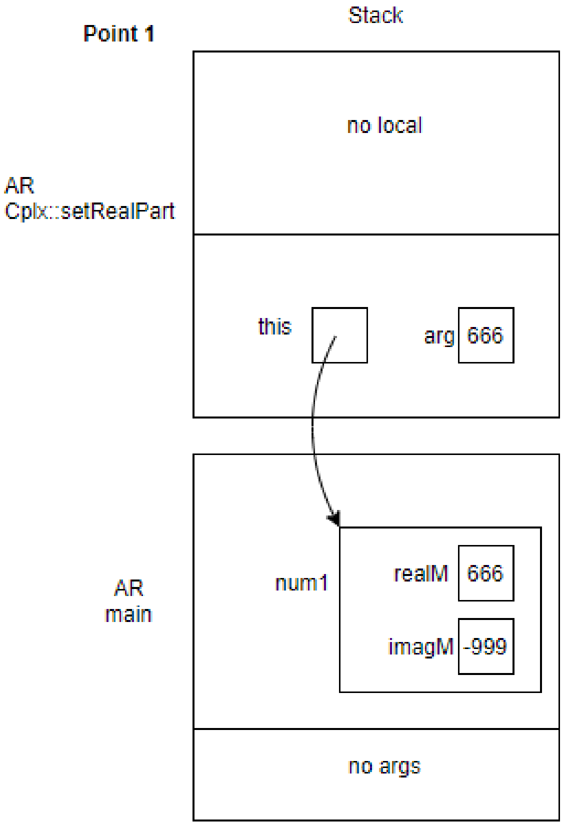
AR Diagrams

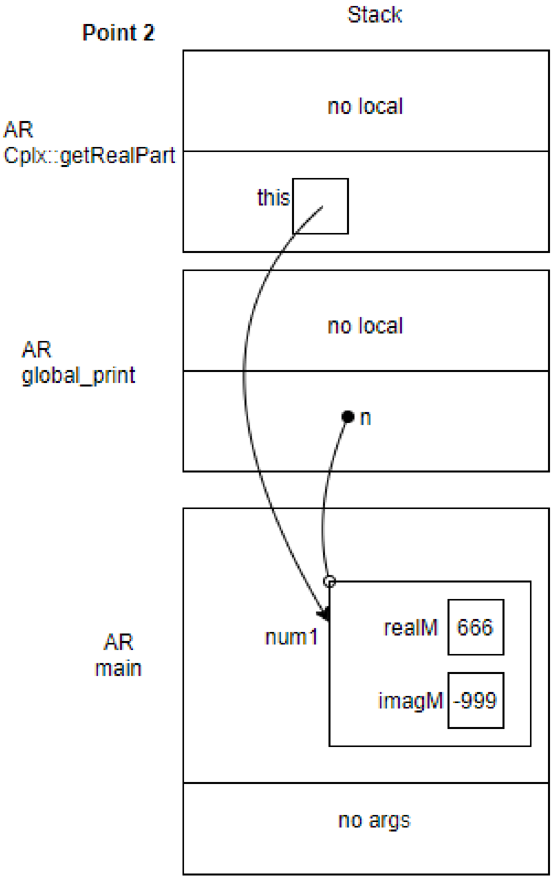




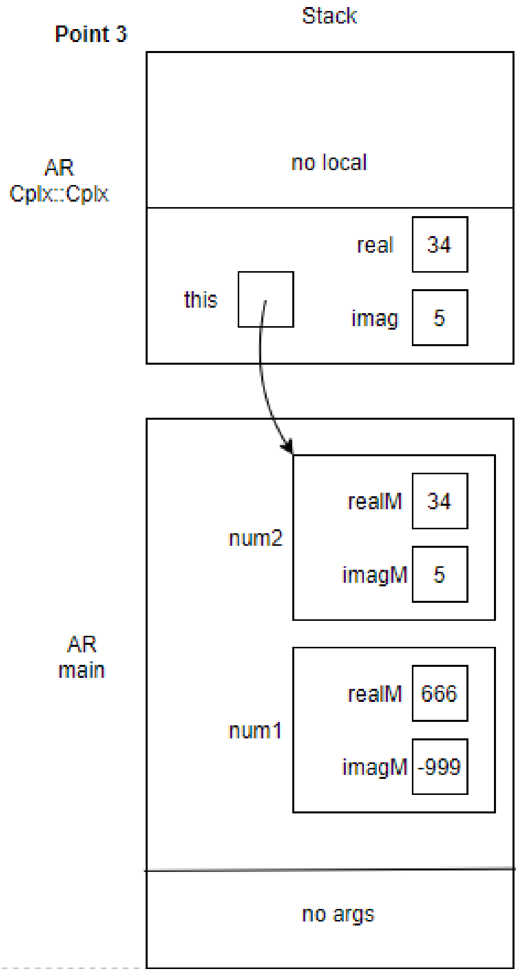
Exercise B

AR Diagrams

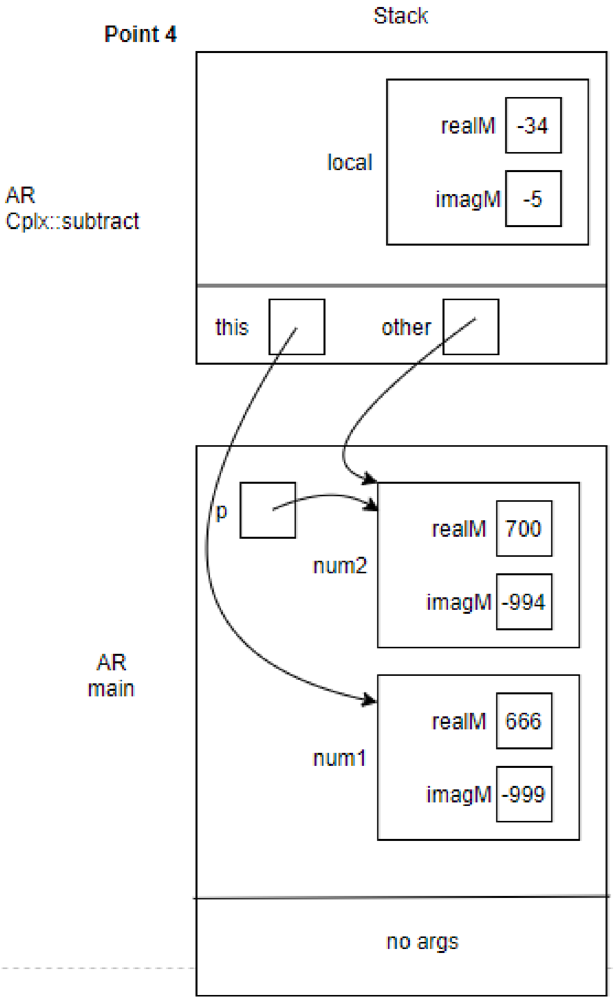




Point 3

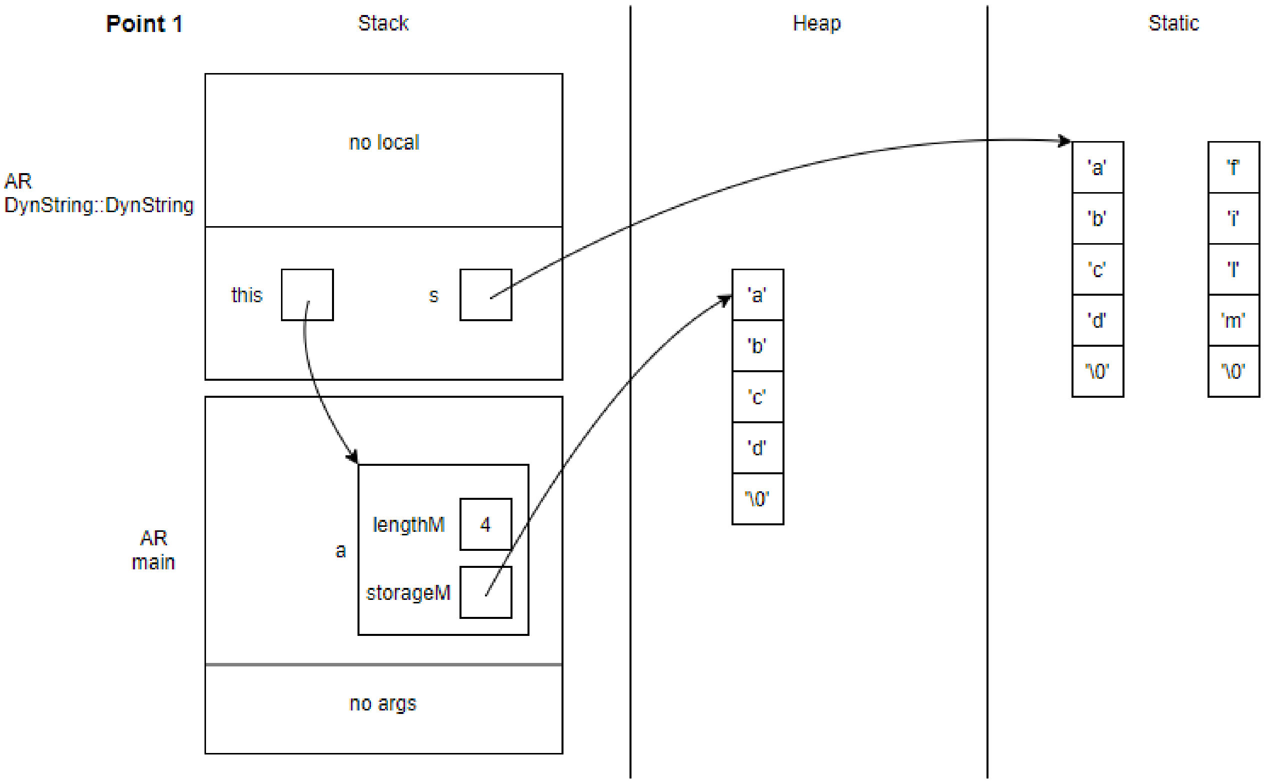


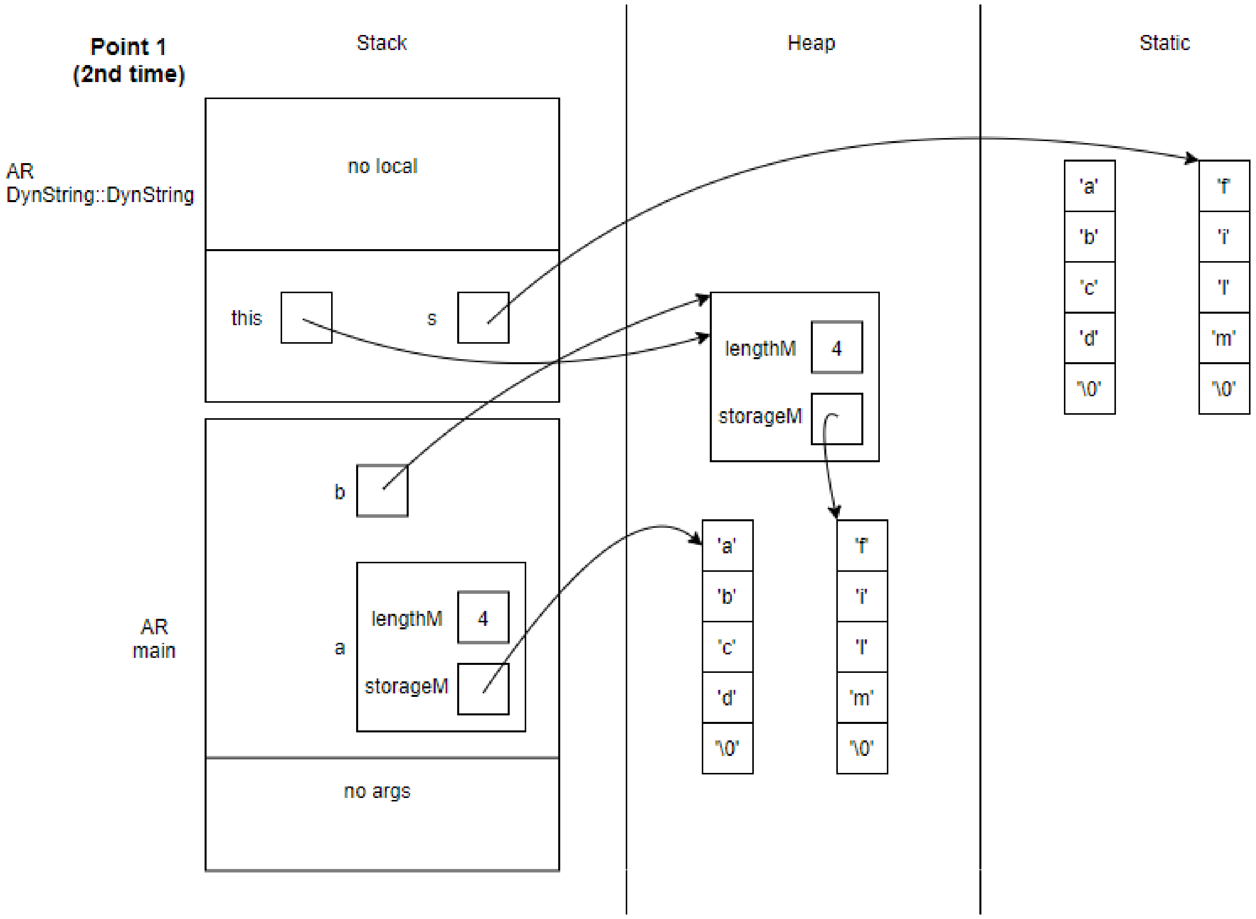
Point 4



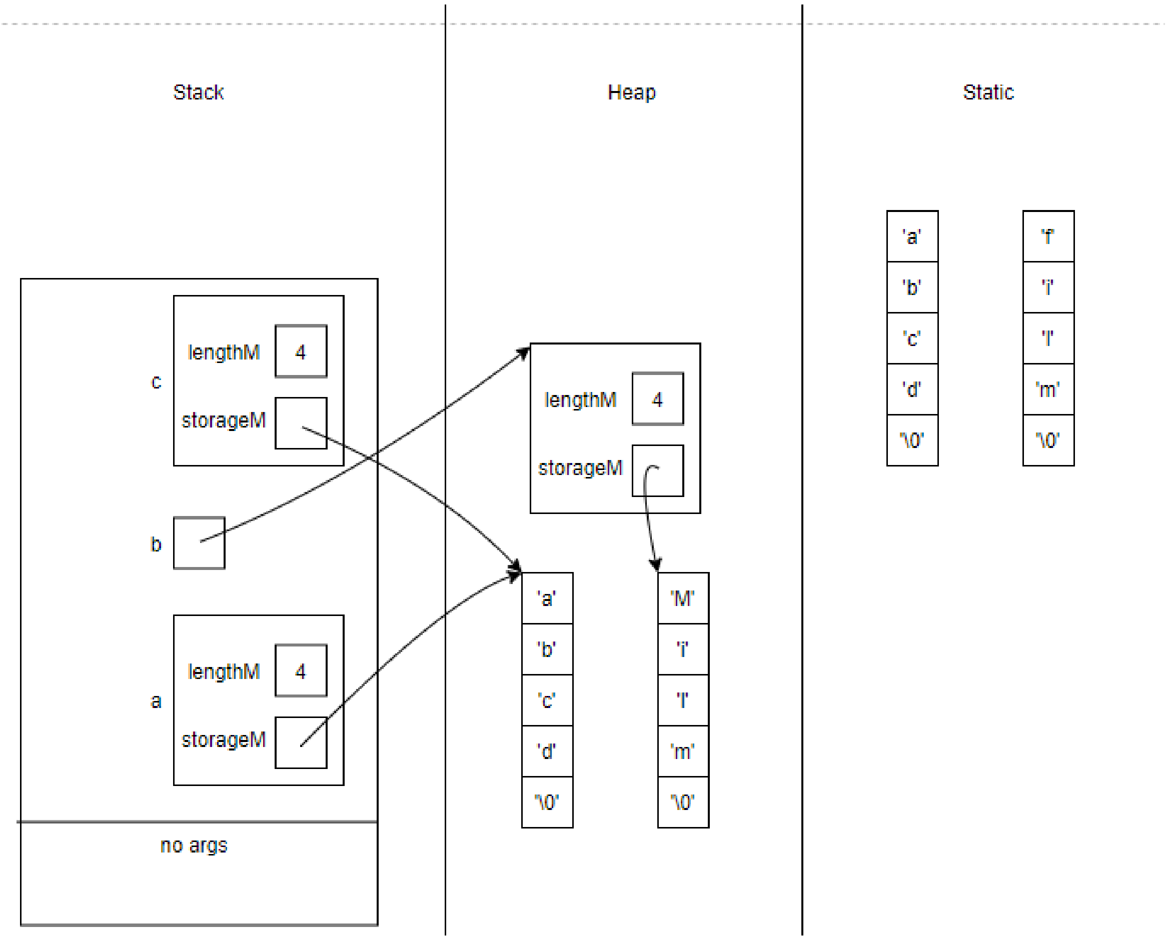
Exercise D Part 1

AR Diagrams

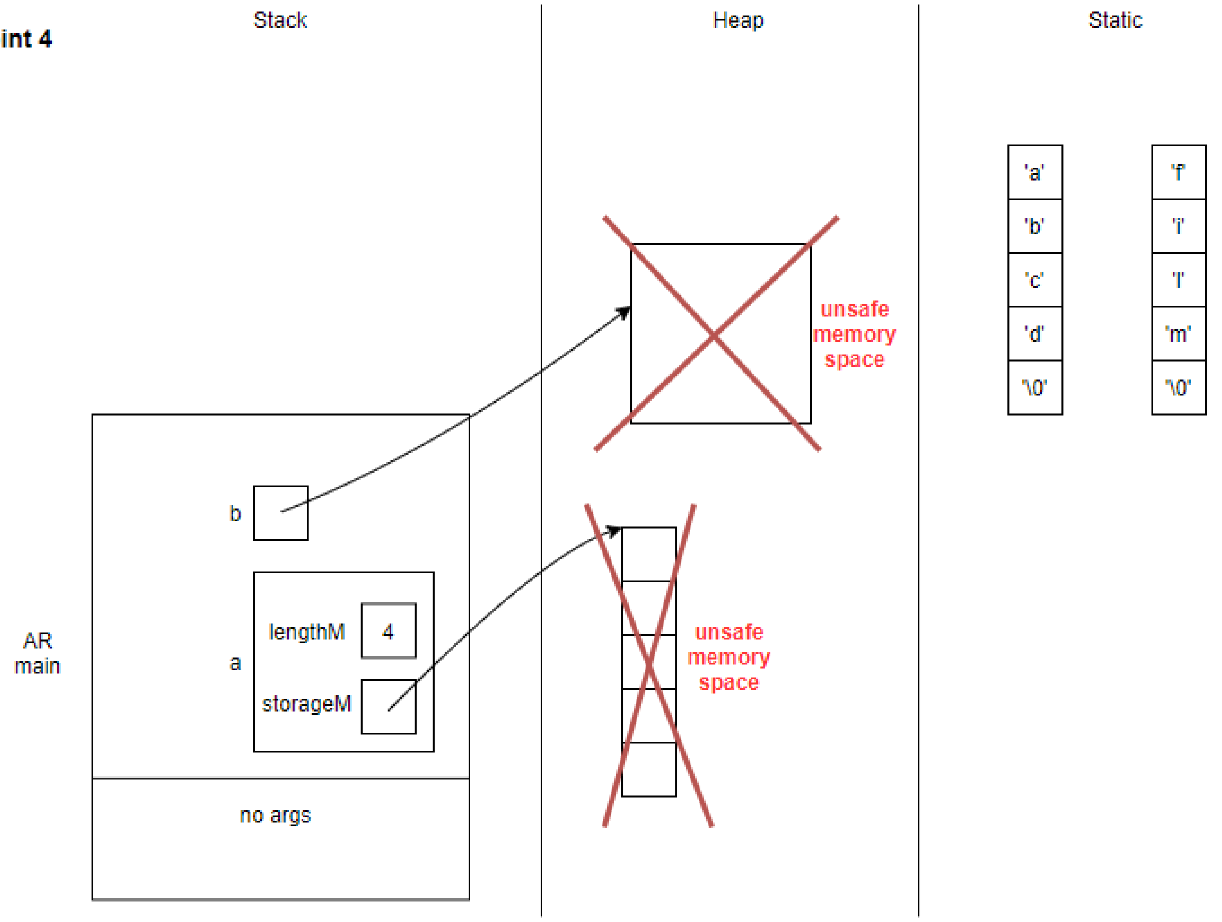




Point 3



Point 4



Questions

1. two times

2. two times
3. three times
4. DynString c was created inside the braces, once control is exiting the braces, it automatically calls the destructor. And since c's storageM is pointing to the same location on the heap as a's storageM due to the shallow copy, it ends up being de-allocated. Once we reach Point 4, b has also been de-allocated, and it is finally at the end of the program after pressing the enter key, the destructor of a is called, but since a's storageM has already been de-allocated, we introduce "undefined behaviour" and we may see an error since you cannot de-allocate already de-allocated memory

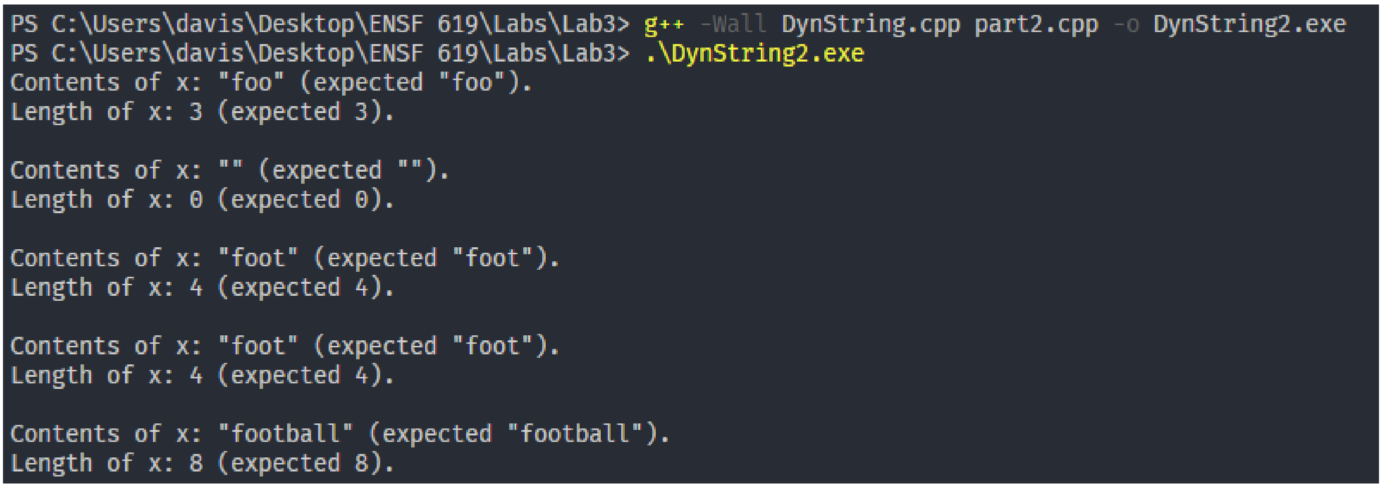
Exercise D Part 2

```cpp
void DynString::append(const DynString& tail)
{
    //create new array of exact length required
    char *appended = new char[lengthM + tail.lengthM + 1];
    assert(appended != NULL);


    //if the string that is getting appended is empty, only copy tail
    if (storageM[0] == '\0') {
        for (int i = 0; i < tail.lengthM; i++){
            appended[i] = tail.storageM[i];
        }
    }
    else {
        int i = 0;
        //copy original chars to new array
        for (int j = 0; j < lengthM; j++) {
            appended[j] = storageM[j];
            i++;
        }

        //append the tail characters
        for (int k = 0; k < tail.lengthM; k++) {
            appended[i] = tail.storageM[k];
            i++;
        }
    }
    //adjusting the new length
    lengthM += tail.lengthM;
    //setting the last character to null
    appended[lengthM] = '\0';
    delete [] storageM;
    storageM = appended;
}
```

Program output



```
PS C:\Users\davis\Desktop\ENSF 619\Labs\Lab3> g++ -Wall DynString.cpp part2.cpp -o DynString2.exe
PS C:\Users\davis\Desktop\ENSF 619\Labs\Lab3> .\DynString2.exe
Contents of x: "foo" (expected "foo").
Length of x: 3 (expected 3).

Contents of x: "" (expected "").
Length of x: 0 (expected 0).

Contents of x: "foot" (expected "foot").
Length of x: 4 (expected 4).

Contents of x: "foot" (expected "foot").
Length of x: 4 (expected 4).

Contents of x: "football" (expected "football").
Length of x: 8 (expected 8).
```