Tufts Online Transcripts (TOT 2.0)

In the first homework assignment you will finish and update the transcript program from lab. The first change is that the implementation for the transcript storage will change from a static array of pointers to transcripts to a dynamic array of transcripts.

TOT2.h - new class definition

```
#ifndef TOT_H
#define TOT_H

#include <iostream>
#include "Transcript.h"

using namespace std;

//initial size of dynamic array
const int INIT_SIZE = 5;

//factor to increase the size of array, we double every time we expand the array
const int RESIZE_FACTOR = 2;

class TOT2
{  public:

    //constructor creates dynamic array of size INIT_SIZE, sets pointers to NULL
    TOT2();

    //same constructor as before, note you may call addTranscript function
    TOT2(Transcript* transcriptList, int listSize);

    //same function as before
    Transcript* getTranscriptCopy(string studentID);

     //new function add a single Transcript to the system, returns bool if it worked
     bool addTranscript(Transcript* t);

    private:
    //dynamic array of pointers to Transcripts
    Transcript** transcripts;

    Transcript* lookUp(string sID);

    //new function to expand dynamic array when it is full
    void expand();

    //count of how many currently stored
    int currentCount;

     //capacity of the dynamic ray
     int currentCapacity;
    };

#endif
```

You will have 6 files in this assignment. We will provide four of these files. They are TOT2.h, TOT2.cpp, main.cpp, TestDat.h, TestDat.cpp, Transcript.h. We are providing the Transcript.h (same as in the lab), the TOT2.h file (same as above), TestDat.h and TestDat.cpp --- new files that declare and define an array of Transcripts. (sample data) for you to use in main.cpp. You will completely write TOT2.cpp and main.cpp. You may use the lab file TOT.cpp as a guide.

TOT2.cpp

You will write all of the necessary functions for this new class with a dynamic array implementation. You may use anything you wrote in lab. You may have additional functions if you like.

main.cpp

You will completely write this to test your code. You will have the necessary include files --- particularly, the new TestDat.h file, the TOT2.h, and Transcript.h files. Read the TestDat.h file and TestDat.cpp file carefully. We have provided an array of data to use. You should test both the default and overloaded constructor as well as the add function which in turn should test your expand function.

getting the files

You may download the files by executing this command from your cs home account. (Logged into the cs department --- ssh linux.cs.tufts.edu or in the lab) In addition to the files necessary to do the assignment you will get Bruce's file generating the test data.

```
cp /comp/15/public_html/files/hw1/* .
```

compiling

To compile you will use a command similar to the lab

g++ -g -Wall - Wextra TOT2.cpp TestDat.cpp main.cpp

Just as we did with the class files, TOT2.h and TOT2.cpp, we *include* the TestDat.h file declaring the array of transcripts in main and *compile* the array definition in TestDat.cpp.

provide

Please provide all 6 files.

```
provide comp15 hw1 TOT2.h TOT2.cpp main.cpp TestDat.h TestDat.cpp Transcript.h
```