Create a project/program named Ch\_7\_HW. The program should use the code from your Ch\_6\_HW as the starting point (just copy/paste your code) and do the following while using the selection structures as detailed below. The parts in green are the new additions/changes:

1. Create a const global of type int that has a value of 10. 10 will be the max number of students we can have and also the max number of grades each student may have. We will limit it here for simplicity.
2. Prompt the user for the number of students grades will be entered for, showing them the acceptable range (1 to 10).
3. Using an if/else statement, check that the number of students is greater than 0 (and less or equal to 10) and display an error message if they are not. Error message should be changed as shown in the screenshots later in this document. Alternatively continue processing if they are good.
   1. Prompt the user for the number of grades that will be entered for each student, showing them the acceptable range (1 to 10).
   2. Using an if/else statement, check that the number of grades to be entered for each student is greater than 0 (and less or equal to 10) and display an error message if it is not. Error message should be changed as shown in the screenshots later in this document. Alternatively continue processing if it is good.
      1. In a while loop prompt the user for the number of points each assignment is worth.
         1. In a function named **pts\_poss**, **pass one integer** for the number of the assignment, get the input in the function, and **return that input to the calling location as a double**.
         2. Using an if/else, check that the number of points returned is greater than or equal to 0. If not don’t increment the loop counter, otherwise increment the counter and add the number of points to an accumulator variable.
      2. After possible points have been entered for all grades print out the total points possible.
      3. In a do/while loop go through the number of students
         1. Nested in the above loop use a for loop to allow each of the grades to be entered for the specific student. The grade should be entered through a call to a function named **ent\_grd** which **receives two integers** (one for student number and one for the grade number) and **returns a double** which is the grade entered.
         2. Using an if/else statement, check that the grade entered is greater than or equal to 0. If it is then add it to an accumulator variable for the student’s grades, otherwise display an error message and decrement the counter variable to pass through the loop again and reenter the grade. At this point you should enter the returned grade (if it is valid) into a 2D array (size 10x10), using the student for the row and the assignment for the column.
         3. When all grades have been entered for that student, print out his/her average by making a call to a function named **find\_avg** which **receives two doubles** (total points the student scored and total points possible) and **returns a double** which is the student’s average.
      4. Call the function named **print\_grades which receives the 2D array holding student grades, an integer representing the number of students, and an integer representing the number of grades for each student**. This function **returns nothing**. It should **use a for loop to go through each student** (row), and **a nested or loop to go through each grade for that student** (column). It should print a message stating the following (example for 2 students with 2 grades each)..also examples of screenshots lower in this document and the exe can be used for as many examples as you would like:

“ Student 1, assignment 1, has a grade of 5”

“ Student 1, assignment 2, has a grade of 4”

“ Student 2, assignment 1, has a grade of 3”

“ Student 2, assignment 2, has a grade of 4”

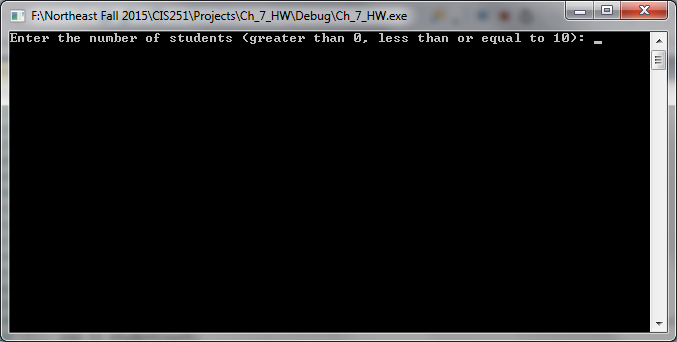
….etc…..

Should continue for all students and grades in the 2D array.

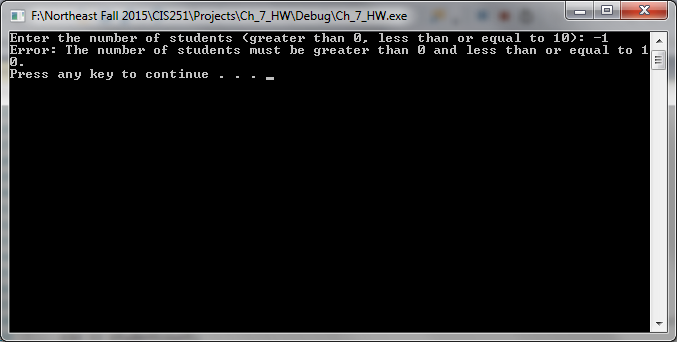
Of course…. A pause at the end so everything is displayed.

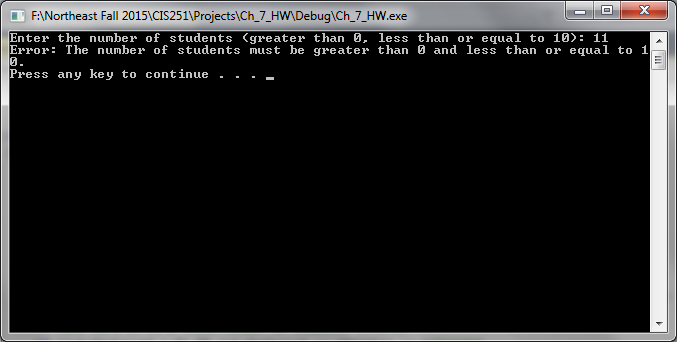
NOTE: I included the exe file so you can run it for examples if you wish to compare to the output from your project. Let me know if you find any problems with it.Now with screen shots examples. All possible situations may not be covered in these screen shots. The .exe file is included to allow you to test examples as needed.

1. Create a const global of type int that has a value of 10. 10 will be the max number of students we can have and also the max number of grades each student may have. We will limit it here for simplicity.
2. Prompt the user for the number of students grades will be entered for, showing them the acceptable range (1 to 10).

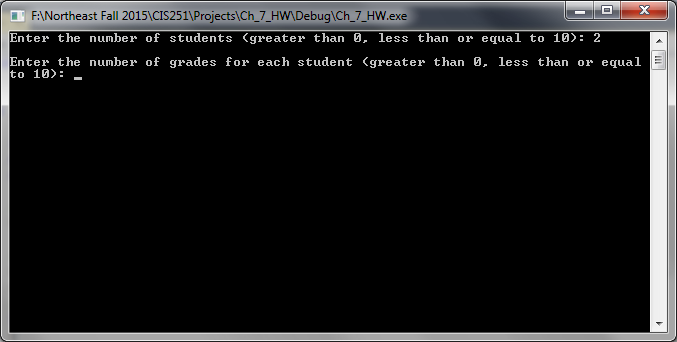


1. Using an if/else statement, check that the number of students is greater than 0 (and less or equal to 10) and display an error message if they are not. Error message should be changed as shown in the screenshots later in this document. Alternatively continue processing if they are good.

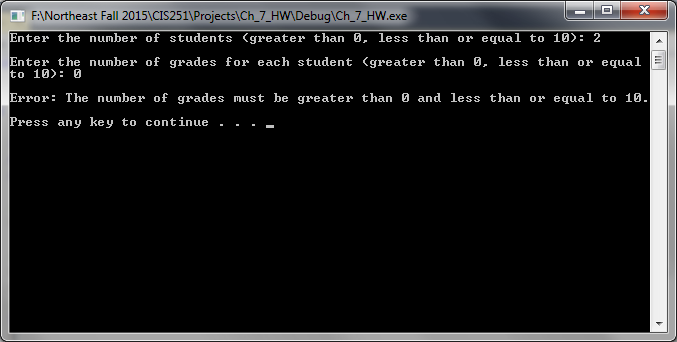


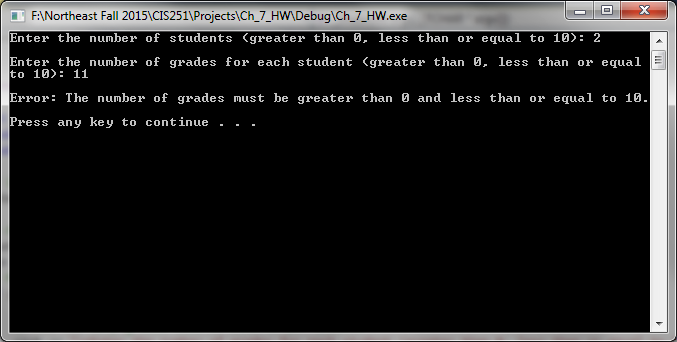


* 1. Prompt the user for the number of grades that will be entered for each student, showing them the acceptable range (1 to 10).

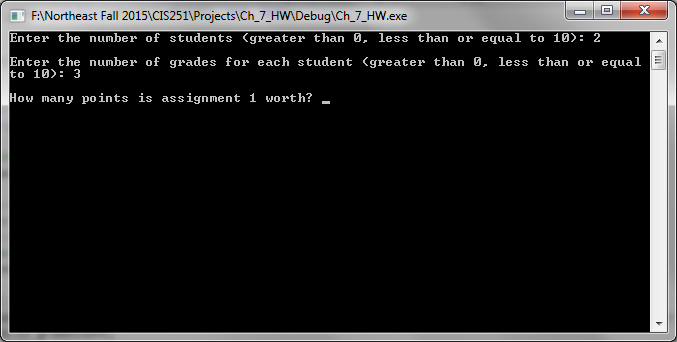


* 1. Using an if/else statement, check that the number of grades to be entered for each student is greater than 0 (and less or equal to 10) and display an error message if it is not. Error message should be changed as shown in the screenshots later in this document. Alternatively continue processing if it is good.



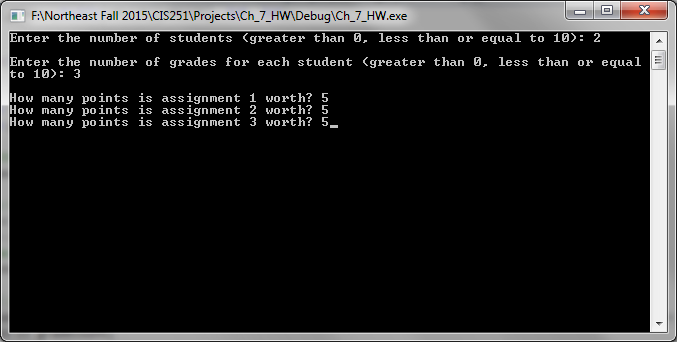


* + 1. In a while loop prompt the user for the number of points each assignment is worth.

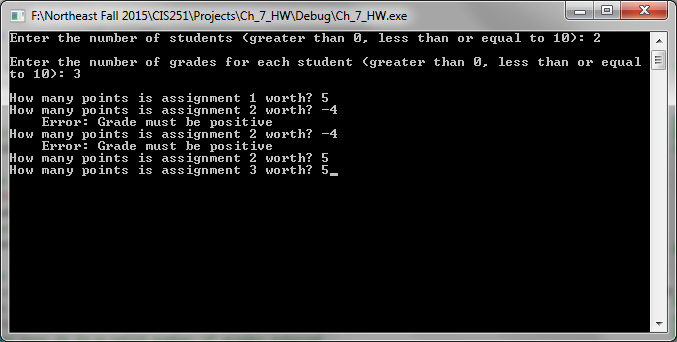


* + - 1. In a function named **pts\_poss**, **pass one integer** for the number of the assignment, get the input in the function, and **return that input to the calling location as a double**.
      2. Using an if/else, check that the number of points returned is greater than or equal to 0. If not don’t increment the loop counter, otherwise increment the counter and add the number of points to an accumulator variable.

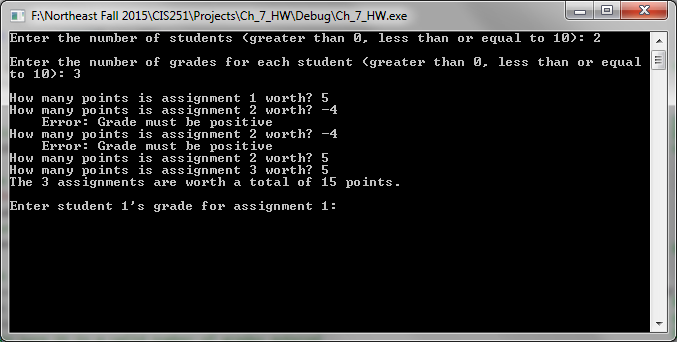
Good:



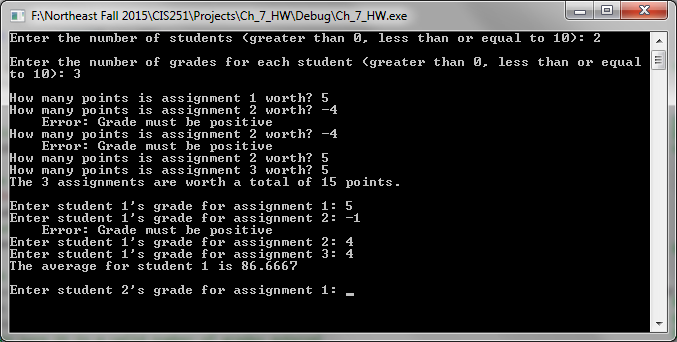
Two bad ones entered (note: don’t exit, just don’t increment the counter!):



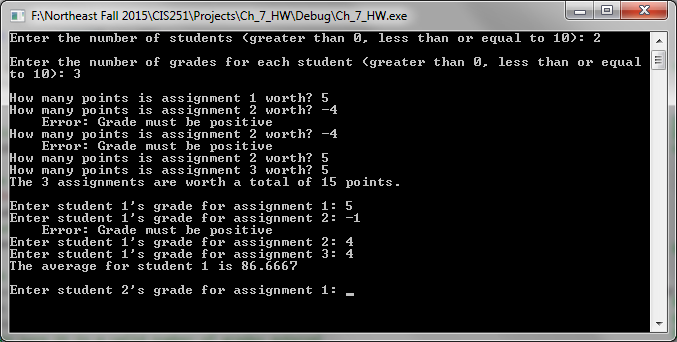
* + 1. After possible points have been entered for all grades print out the total points possible.



* + 1. In a do/while loop go through the number of students
       1. Nested in the above loop use a for loop to allow each of the grades to be entered for the specific student. The grade should be entered through a call to a function named **ent\_grd** which **receives two integers** (one for student number and one for the grade number) and **returns a double** which is the grade entered.
       2. Using an if/else statement, check that the grade entered is greater than or equal to 0. If it is then add it to an accumulator variable for the student’s grades, otherwise display an error message and decrement the counter variable to pass through the loop again and reenter the grade. At this point you should enter the returned grade (if it is valid) into a 2D array (size 10x10), using the student for the row and the assignment for the column.



* + - 1. When all grades have been entered for that student, print out his/her average by making a call to a function named **find\_avg** which **receives two doubles** (total points the student scored and total points possible) and **returns a double** which is the student’s average.



* + 1. Call the function named **print\_grades which receives the 2D array holding student grades, an integer representing the number of students, and an integer representing the number of grades for each student**. This function **returns nothing**. It should **use a for loop to go through each student** (row), and **a nested or loop to go through each grade for that student** (column). It should print a message stating the following (example for 2 students with 2 grades each)..also examples of screenshots lower in this document and the exe can be used for as many examples as you would like:

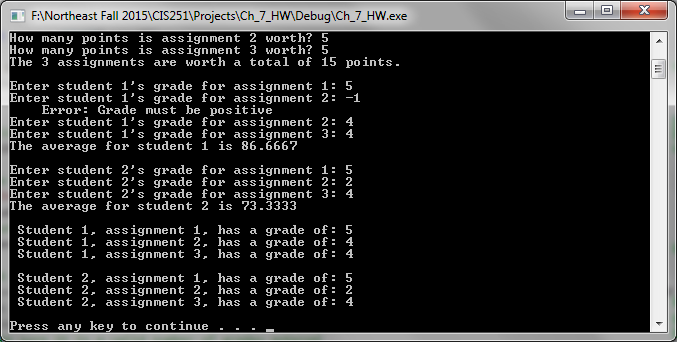
“ Student 1, assignment 1, has a grade of 5”

“ Student 1, assignment 2, has a grade of 4”

“ Student 2, assignment 1, has a grade of 3”

“ Student 2, assignment 2, has a grade of 4”

Should continue for all students and grades in the 2D array.



Of course…. A pause at the end so everything is displayed.