The purpose of this assignment is to demonstrate the benefits of dynamic programming. The task for the program to perform is to take an input text file containing a triangle of integers and output the maximum sum of the integers when taking a path from the top of the triangle to the bottom. The problem with a brute force approach is that the number of paths the program would need to traverse is approximately 2^(x-1) where x is the number of rows in the text file. In the case of “Triangle.txt”, which contains 100 rows, the number of traversals necessary is approximately 6.33 x 10^29. Instead this program offers a different approach where it starts at the second to the bottom row and compares the 2 integers directly beneath each integer in the row we’re working in. It takes the largest of these two integers and adds it to the integer above it. If this process is repeated for each integer in the row and each row in the text file, the maximum sum of the triangle will be calculated at the top of the triangle; thus solving the problem with greater time efficiency than would be possible with a brute force method. The program’s implementation can be found in the file “MaximumPath.cs”.

