1. Write a function `bool IsOddDigit( char ch );` that takes a character and returns true if the character is one of the odd digits \{ '1', '3', '5', '7', '9' \}, otherwise false. Write a main() program that calls `IsOddDigit()` with various characters, and uses the return value from `IsOddDigit()` to decide to print one of two messages, that the character is an odd digit or that the character is not an odd digit. (20 points)

   The function `IsOddDigit()` may neither print output nor accept input. You may make none of the decisions about the character's properties outside of `IsOddDigit()`.

   Test with at least all the digits '0' through '9', the letters 'a', 'b' and 'c' (both lower and upper case), and several other characters such as punctuation marks and other special characters. You may use any method you want to generate characters to send to `IsOddDigit()`. You do not have to prompt for the characters.

   Hints: "takes" and "returns" are technical terms in computer science -- a function "takes" its parameters (or arguments) and "returns" its return type or return value. This is a tiny function. The entire program can be written in no more than about 30 lines, including notes. (20 points)

2. Recall the parallelogram assignment from assignment 6. You will now break the program into functions for the various tasks. You will have the following functions other than main():

   // get the number of rows from the user, forcing legal range and // then making sure it is odd, then return it to the caller
   int GetNRows( void );

   // draw the shape, using helper functions described below
   void DrawShape( int nRows );

   // calculate number of spaces on a row, // given number of rows and current row number
   int CalcNSpaces( int nRows, int row );

   // calculate number of stars on a row, // given number of rows and current row number
   int CalcNStars( int nRows, int row );

   // print n copies of character c on the current row
   void PrintChars( int n, char c );

   **YOU MAY NOT USE ANY GLOBAL VARIABLES: ALL VARIABLES MUST BE DECLARED WITHIN THE APPROPRIATE FUNCTIONS AND PASSED AROUND AS NEEDED.** You may not add any parameters to any of these functions. The function must do ONLY the task described in the note above the prototype. If you did the parallelogram correctly the first time, this program will be easy. If not, you **MUST** organize the code along these lines. Your main() function will be quite short. (40 points)