THIS IS AN EXAMPLE OF HOW a *second level* PROGRAM SUBMISSION SHOULD LOOK. IT ILLUSTRATES CORRECT CODING STYLE, CODING FORMAT, NAMING CONVENTIONS, AND DOCUMENTATION.

/* PROGRAM Stars */
/*
Name: Gordon Goodguy
User Name: goodge
Course: CS 3100, Data Structures & Algorithms
Instructor: Ed Lamie
Date: October 1, 2001
INPUT:
The user of this program inputs a 'y' for yes or a 'n' for no.
OUTPUT:
This program prints a little message and then repeatedly asks the user if he wants to see a pattern. When the user answers in the affirmative, it prints the pattern and asks the question again. When the user answers with 'n', the program stops.
PRECONDITIONS and POSTCONDITIONS: None.
*/

#include <iostream.h>

/*
FUNCTION NAME: PrintMessage ;
INPUT: none.
OUTPUT: a message to the user of this program.
PRECONDITIONS: output set to start on a new line.
POSTCONDITIONS: output set to start on a new line.
CALLERS: the main program
CALLEES: none.
*/

void PrintMessage()
{
  cout << endl ;
  cout << "F. Scott Fitzgerald wrote a story about a diamond" << endl ;
  cout << "as big as the Ritz Hotel in New York. I hope you" << endl ;
  cout << "like the little diamond that this program writes," << endl ;
  cout << "even though it is not a real one, and certainly not" << endl ;
  cout << "anywhere near as large as the Ritz Hotel." << endl ;
  cout << endl ;
}

/*
FUNCTION NAME MakeStarRow ;
INPUT: the number of stars to be made (this is the parameter amountMP).
OUTPUT: a row of asterisks centered in a field of 75 characters.
PRECONDITIONS: output set to start on a new line.
*/
void MakeStarRow(int amountMP)
{
    /* stub action */
    cout << "Here are " << amountMP << " stars." << endl;
}

void MakePattern(int widthMP)
{
    int amount;
    for (amount=1; amount<=widthMP; amount++) MakeStarRow(amount);
    for (amount=widthMP-1; amount>=1; amount--) MakeStarRow(amount);
}

main()
{
    char response;
    PrintMessage();
    do{
        cout << "Would you like to see a pattern? " ;
        cout << "Answer y or n: " ;
        cin >> response;
        if (response == 'y') MakePattern(15);
    }while(response != 'n');
}