```
//
//   Name              John Keller
//   Recitation TA     Jason Zietz
//   Assignment #      3
//   Problem #         1
//

#include <iostream>

using namespace std;

void madLibs(void);

int main() {
    // Declare the variables
    bool inSession = true;
    string play = "";

    // Loop the function so when the user is done playing, it returns them to the
        ↪ beginning
    while (inSession) {
        // Retrieve user input
        cout << "Do␣you␣want␣to␣play␣a␣game?␣(y)␣or␣(n)";
        cin >> play;
        if (play == "y") {
            // User said they wanted to play, so let's run the madLibs() function
            madLibs();
        } else if (play == "n") {
            cout << "good␣bye" << endl;
            inSession = false; // This boolean is what kicks the user out of the
                ↪ loop, ending the program
        }
    }
}

void madLibs(void) {
    // Declare our story string, including the placeholders
    string story = "In␣the␣book␣War␣of␣the␣<PLURAL␣NOUN>,␣the␣main␣character␣is␣an
        ↪ ␣anonymous␣<OCCUPATION>␣who␣records␣the␣arrival␣of␣the␣<ANIMAL>s␣in␣<
        ↪ PLACE>␣--␣Needless␣to␣say,␣havoc␣reigns␣as␣the␣<ANIMAL>s␣continue␣to␣<
        ↪ VERB>␣everything␣in␣sight,␣until␣they␣are␣killed␣by␣the␣common␣<SINGULAR
        ↪ ␣NOUN>.";

    // Declare the titles and items to replace in story (using an array)
    string parts[6][2] = {
        {"a␣plural␣noun","<PLURAL␣NOUN>"},
        {"a␣singular␣noun","<SINGULAR␣NOUN>"},
        {"an␣occupation","<OCCUPATION>"},
        {"an␣animal␣name","<ANIMAL>"},
        {"a␣place","<PLACE>"},
        {"a␣verb","<VERB>"}};

    // Go through the arrays, retrieving a user input and replacing the
        ↪ placeholder string
    for (auto & part : parts) {
        // Declare our variables
        string value = "";
        string title = part[0];
        string placeholder = part[1];
        // Retrieve the user input
        cout << "Enter␣" << title << ":␣";
        cin >> value;

        // Let's determine if the placeholder exists within the story, if so,
           ↪ replace it with the user's value
        // And let's put it in a loop, so we can catch repeating placeholders
        while (story.find(placeholder)!=string::npos) {
            story.replace(story.find(placeholder),placeholder.length(),value);
            // Note: .replace(startPosition, length, newString)
            //       .find(string) returns the start index of the placeholder
               ↪ string
            //       .length() does the obvious and returns the length
        }
    }

    // Print the final string
    cout << story << endl;

    return;
}
```

```
//
//  Name            John Keller
//  Recitation TA   Jason Zietz
//  Assignment #    3
//  Problem #       2
//

#include <iostream>

using namespace std;

void printEnergy(double A, double r, double H, double PR);
double energyCalculator(double A, double r, double H, double PR);
int calculateNumberHousesSupported(double array_avg,double household_avg);

int main() {
    // Declare variables
    double A = 0;
    double H = 0;
    double average = 0;
    // Predefined variables:
    double r = 0.10;
    double PR = 0.75;

    // Retrieve the inputs for A and H
    cout << "A:␣";
    cin >> A;
    cout << "H:␣";
    cin >> H;

    // Part A
    double calculated_energy = energyCalculator(A, r, H, PR);
    cout << "The␣average␣annual␣solar␣energy␣production␣is␣" << calculated_energy
       ↪ << "kWh." << endl;

    // Part C
    cout << "Enter␣household␣average:␣";
    cin >> average;
    cout << calculateNumberHousesSupported(calculated_energy,average) << "␣houses␣
       ↪ can␣be␣supported." << endl;

    // Part B
    while (r < 0.36) {
        printEnergy(A, r, H, PR);
        r = r + 0.05; // Add 0.05 to the previous r value
    }
}

void printEnergy(double A, double r, double H, double PR) {
    // Prints a human-readable version of the calculated energy data
    cout << "The␣average␣annual␣solar␣energy␣for␣an␣efficiency␣of␣" <<  r << "␣is␣
       ↪ " <<  energyCalculator(A, r, H, PR)  << "␣kWh." << endl;
}

double energyCalculator(double A, double r, double H, double PR){
    /* Calculates energy data using the equation E = A * r * H * PR
     *  A is area of the solar array in meter square
     *  r is the solar panel efficiency in percentage
     *  H is the average solar radiation in kWh/m^2/year PR is the performance
        ↪ ratio (coefficient of loss)
     *      - usually between 0.5 and 0.9, with a default of 0.75
     */
    return A * r * H * PR;
}

int calculateNumberHousesSupported(double array_avg, double household_avg) {
    // Perform calculations and return values
    int number_of_houses = array_avg / household_avg;
    return number_of_houses;
}
```