```
// CSCI 301
// Computer Science 2

// File: format.cxx

// This program reads an input file of text and writes an output file of the
// same text, formatted into lines no longer than a maximum length. The
// names of the input and output files and the maximum line length are
// read from the terminal. Functions open the files, and continue prompting
// for file names until names are entered than can be successfully opened.
// Another function reads the maximum line length, which must fall within
// bounds set by two program constants.

// The program reads and writes words from the input file one at a time.
// It keeps track of the length of the current line so far; if the next
// word would cause that line to exceed the maximum length, the program
// terminates that line, writes the word on the next line, and resets the
// line length. The program writes a blank after each word, except perhaps
// the last word on a line. A word is a string of contiguous non-blank
// characters, and we assume that no input word is longer than the input line
// length set for the run.

#include
#include
#include
#include
using namespace std;

const int MIN = 30;                    // Minimum line length
const int MAX = 80;                    // Maximum line length

typedef char string[MAX+1];

void open_input_file ( ifstream& in_f );
// Opens for input a file named from the terminal.
// Postcondition: A file stream has been opened for input.

void open_output_file ( ofstream& out_f );
// Opens for output a file named from the terminal.
// Postcondition: A file stream has been opened for output.

int read_int ( int small, int large );
// Reads an input value within specified bounds.
// Precondition: small and large are positive integers, with small <= large.
// Postcondition: The function returns a value in [small,large] entered from
//   the terminal.

int main()
{
  ifstream in_file;            // The input file stream
  ofstream out_file;           // The output file stream
  int max_length;              // Maximum line length
  string s;                    // Each string read in and printed out
  int s_len;                   // The length of the string s
  int line_len;                // The length of the current output line so
far
```

```
  open_input_file(in_file);    // Open the input file.
  open_output_file(out_file);  // Open the output file.
  max_length = read_int(MIN,MAX); // Read the maximum line length.

  line_len = 0;                // Initially, the line length is zero.
  in_file >> s;                // Read from the input file.
  while ( ! in_file.eof() )    // Are we done yet?
  {
    s_len = strlen(s);         // Identify the string's length.
    if ( line_len + s_len <= max_length )  // If there is room on the line
...
    {
      out_file << s;           // Write to the output file.
      line_len = line_len + s_len;  // Increment the line length.
    }
    else                       // Start a new line.
    {
      out_file << endl << s;   // Write to the output file.
      line_len = s_len;        // Reset the line length.
    }

    if ( line_len < max_length )  // If there is room for a blank ...
    {
      out_file << ' ';         // Write to the output file.
      ++line_len;
    }
    in_file >> s;              // Read from the input file.
  }
  out_file << endl;            // Write to the output file.
  in_file.close();             // Close the input file.
  out_file.close();            // Close the output file.

  return EXIT_SUCCESS;
}

void open_input_file ( ifstream &in_f )
{
  char input_file_name[80];

  do
  { in_f.clear();
    cout << "Enter input file name: ";
    cin >> input_file_name;
    in_f.open(input_file_name);
  } while ( in_f.fail() );
}

void open_output_file ( ofstream &out_f )
{
  char output_file_name[80];

  do
  { out_f.clear();
    cout << "Enter output file name: ";
    cin >> output_file_name;
    out_f.open(output_file_name);
  } while ( out_f.fail() );
```

```
}

int read_int ( int small, int large )
{
  int value;

  do
  { cout << "Enter an integer value between " << setw(1) << small
         << " and " << setw(1) << large << ": ";
    cin >> value;
  } while ( value < small || value > large );
  return value;
}
```