Introduction to C - Programming Assignment #6

Due date: December 2, 2011, 11:59pm

Problem: Guest List (guest_struct.c)
Rewrite the guest list problem from program #5 using the following struct/typdef:

```
typedef struct family {
    char first[MAX_LEN];
    char last[MAX_LEN];
    int num_people;
    int priority;
} family_type;
```

Your solution should use a single array of family_type.

You are still required to use multiple functions.

Deliverables
The source file, *guest_struct.c*, containing your solution to the problem.

Restrictions
Although you may use other compilers, your program must compile in gcc and run in the Code::Blocks environment. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details
Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility to gcc in Code::Blocks. If your program does not compile in this environment, you will get a **sizable** deduction from your grade.

Extra Credit 1(20pts)

Write a recursive function to compute the sum of squares of a range of numbers. Use the following function prototype

```
int SumSquares(int m, int n);
```

Note that m < n. As an example, a call to SumSquares(5,10) would return a result of 355 since 5*5 + 6*6 + 7*7 + 8*8 + 9*9 + 10*10 = 355. Also note you must use recursion to get credit.

Extra Credit 2 (20pts)

Write a recursive function that reverses a subset of a string. Use the following function prototype

```
void ReverseString(char *s, int m, int n);
```

where m an n are indices into the string. Note that m < n. As an example, given a string s = “abcdefg” and a call to ReverseString(s, 1,4) would change the string to “aedcbg”. Note that you must use recursion to get credit.

For both programs, you should repeatedly ask the user for the parameters to the functions and so you can easily test your solutions. Enter a -1 to stop the iteration to quit the program.

Deliverables

The source file `sumsquares.c` should be submitted for the first extra credit problem and `reversestring.c` should be submitted for the second extra credit problem.