CSE 224 HW #4 Modular Coding

For this assignment, you'll be implementing a fairly simple C program, but:

1.you're required to implement it using arrays;

2.you're required to break your program into at least 4 separate .c files and one .h file; and

3.you must create a Makefile for building your executable program.

PROGRAM BEHAVIOR

The program does the following:

•repeatedly prompt the user to enter a number, or to just hit Enter if they are done entering numbers;

•when the user is done entering numbers, display a single line that looks like:

$num_1*num_2*num_3*...*num_n=product$

For example, if the user enters the following numbers: 2 5 3

then your program should display a line that says

2*5*3=30

IMPLEMENTATION

•Your program must be written in standard C; it should compile with "gcc" and be runnable on the CTEC server.

•You should read each number using the fgets() function, and convert from a string to an integer using atoi, sscanf, or some other method of your choice.

•You should store each number you read into an array.

•When the user finishes entering numbers, multiply the numbers you've stored in the array and display your output as described above.

MODULARIZATION

Implement your program as a set of four functions:

•int read_array(int *array) which reads the user's input lines, converts them into integers, and stores them into consecutive locations in the array (which is passed as the argument). This function should return the total number of integers that were entered.

•int mul_array(int *array,int length) which accepts an array and an integer containing the length of the array. The function returns the product of the elements in the array.

•void print_array(int *array,int length) which accepts an array and its length, and prints the output described above. print_array() should call mul_array() to determine the sum

•a main program, which calls read_array and print_array

Your array should be declared in your main program, and will have a fixed size; but that size should be defined in an include (.h) file using a #define statement. Each of your functions should include that file, in order to know the maximum numbers of numbers allowed.

Please name your main program main.c; your other functions read_array.c, mul_array.c and print_array.c and your include file sum.h

Finally, include a Makefile for creating an executable program named "main"

ERROR CHECKING

When you convert the user's input to an integer, check the that the conversion was successful, and give an error message if not (and handle things properly in that case). Also, **make sure the user does not enter too many numbers**. If the array can only hold 10 numbers, and the user tries to enter 11 numbers, inform them that that is not allowed, and continue processing with just the first 10 numbers they entered.

SUBMISSION

Collect all your files - .c, .h and Makefile - into a single archive using tar; then gzip it, and change the extension to .tgz; then upload your .tgz file via Canvas by the due date.