This assignment is to allow you to work with Makefiles and Conditional Compilation.

Makefiles: For this assignment (and pretty much all others after this assignment) I want you to use a Makefile for compilation purposes. This should be a pretty standard Makefile with targets for "all" and "clean". You should also have a target for "debug", which will turn on the debug flag during compilation and trigger your debug statements during conditional compilation. You don't need to generate .o files for this assignment, so you don't need to include that in the makefile. Make sure that you "cat" the makefile when making the script, and also that you include the makefile when submit your code.

The assignment: For this assignment, you are going to write a fairly simple math testing application where a simple math equation is generated and you provide the answer. Write four separate functions - add(), subtract(), multiple(), and divide() - where each function will generate two random numbers to be used in the equation. The user is then asked for the answer, which is checked and marked as "correct" or "incorrect". The function should return a 0 for incorrect and a 1 for correct, and the main() function should keep track of the number of correct and incorrect answers for each type of equation. The program should randomly ask the users questions from the four question types until the user enters in a -1111 as the answer, in which case the function should return a -1, exit the main loop, print out your "stats" and end the program. This program should have full error-checking and adhere to all good coding practices.

Note: You can generate a pseudo-random number by calling the function rand(). This is usually done by starting the random number generator with a seed, and then calling rand() to generate the random numbers. You seed the random number generator by calling srand() and passing it a number to seed the random number generator. srand() is only called once in a program, and then rand() is used to get numbers.

One problem with seeding a random number generator is that if you give it the same seed, it generates the same random numbers. For example, when I run the code:

```
srand(50);

printf("%d\n", rand());

printf("%d\n", rand());

printf("%d\n", rand());
```

I always get the same three numbers:

1920540202

1350000510

223544838

This is because I am using a pseudo-random number generator. This is often "fixed" by picking a different seed each time we run the program. An easy way to make sure that you get a different seed each time you run the program is to do the following:

```
#include <time.h>
```

srand(time(NULL));

The time(null) will return (as an int) the current time in seconds, and that can be used as a seed for the random number generator.

Added note: You can limit your random numbers to between 0 and 12. For the division question, you may want to limit your denominator numbers to smaller than the numerator, and you may also want to limit the precision on the number of significant digits that you check. Note that dividing by 0 is generally frowned upon, so you can take whatever steps you need to make sure that you never divide by zero.