CSCI 2240
Program #1

Math Quiz

Concepts Covered

- Basic C structure
- Input/Output
- Control Structures
    - Decision
    - Repetition
- User defined functions
- rand()

Your task is to write a C program that randomly generates, conducts, and scores a simple arithmetic quiz based on certain user parameters. The arithmetic quiz will ask questions dealing with simple addition, subtraction, multiplication, and division. The user will be asked to enter the number of questions for the quiz, between 1 and 20 (be sure to verify and re-prompt if necessary) and a difficulty level between 1 and 4 (also verify).

The difficulty will determine the range of numbers that are randomly generated for the questions:

Difficulty	Range (all integers)
1	1 – 10
2	1 – 50
3	1 – 100
4	-100 – 100

So a quiz with a difficulty of three might have a problem like “75+4” but would not have a problem such as “-75+4”, since negative numbers are only included at difficulty level four.

Once these parameters are set, the quiz will begin, providing the user with simple arithmetic statements, where the operations (+, -, *, /) and operands are randomly generated using rand().Make sure the quiz is different every time!

In the case of division, do not allow the second operand to be zero. Note that in math division of integer number may result in a floating number; but in C division of two integers results in a integer, the remainder is thrown away. Since this assignment focuses on function usage and control flow, to keep things simple it is acceptable to implement just integer division.

For each question the user is prompted to enter their answer and will be told if they are correct or incorrect. There must be three correct and incorrect responses, and the one used

will be randomly selected. If you take a look at the example below, you will see that "Nice!", "Good job!" and "You're right!" are the three randomly selected responses for correct answers, and there are three more for incorrect answers. If the user answered incorrectly, the program also prints the correct answer.

After all the questions are given and answered, the score is provided by stating the number of questions correct/number of questions asked.

For the most part how you write the program solution is up to you; however it must include at least these functions (feel free to add more):

generate_question() – Used to randomly generate and output a random question based upon a difficulty which is passed as a parameter. Returns the answer of the question.

answer_question() – Prompts user for an answer, checks if it is correct based on the correct answer, which is passed to the function as a parameter. Returns an indicator of whether or not the answer was correct.

print_response() – Outputs a response to the user's answer based upon a passed parameter which indicates whether the answer was correct or incorrect. The function randomly selects one of the three responses, either from the correct responses or the incorrect responses.

Example Runs:

Run 1:

```
How many questions for this test (1 - 20)? 5
Select difficulty (1 - 4): 2
Question 1: 23 / 29 =
Enter Answer: 0
Nice!
Question 2: 47 * 8 =
Enter Answer: 376
Good Job!
Question 3: 50 - 5 =
Enter Answer: 42
Sorry!
The correct answer was 45
Question 4: 7 + 24 =
Enter Answer: 31
Good Job!
Question 5: 32 / 43 =
Enter Answer: 0
Nice!
Your score was 4/5
```

Run 2:

```
How many questions for this test (1 - 20)? 25
How many questions for this test (1 - 20)? 4
Select difficulty (1 - 4): -6
Select difficulty (1 - 4): 1
Question 1: 2 - 7 =
Enter Answer: -5
Nice!
Question 2: 4 + 4 =
Enter Answer: 9
Sorry!
The correct answer was 8
Question 3: 10 / 9 =
Enter Answer: 1
You're right!
Question 4: 7 / 1 =
Enter Answer: 7
Good Job!
Your score was 3/4
```

Note: After the programs are collected, any compiled program (executable) in that directory is deleted and the program is recompiled for testing. Specifically, make sure you compile the program with the same options that are used for grading:

```
gcc -Wall –ansi –pedantic quiz.c
```

or whatever the file name is. If the compiler generates any errors or warnings you will be penalized.

Please include a script file used to compile your program. You can find examples in my lectures posted on Blackboard.