

For homework #2 you are to write a Math Tutor program for small children. Your program's interaction should look something like: (Student input shown in **bold**.)

```

Welcome to the Math Tutor

Enter your name: Bob<Enter>
Enter your grade (1, 2, or 3): 3<Enter>

Directions: Please answer the following 5 problems Bob.

Problem:  395
         + 137
         ----
Answer:   422<Enter>

Sorry Bob, but your answer is incorrect.  The correct answer is 532.

.
.
.

Problem:  427
         + 344
         ----
Answer:   771<Enter>

Great Bob, your answer is correct!!!

===== Summary =====
Bye Bob!  You answered 4 out of 5 problems correct.

```

Your program should randomly select the difficulty of the problems according to the following table.

Grade Level	Range of the Numbers
1 (first grade)	0 to 9
2 (second grade)	10 to 99
3 (third grade)	100 to 999

Your program also needs to:

- be designed using the functional-decomposition (top-down) design **before you start to write code!** You will need to turn in a design document (see below for details).
- name your program `mathTutor.py` in a folder `hw2` containing it,
- align the numbers to be added,
- check for a valid grade selection. Anything other than '1', '2', or '3' should print an appropriate error message and quit the program.

Extra credit features: (You don't need to all features to get partial extra credit)

- Randomly pick the arithmetic operation based on the grade level. First grade should only be addition. Second grade can be addition or subtraction, but if subtraction is to be performed the "upper" number should be larger than the "lower" number. Third grade should allow addition, subtraction, and multiplication. Third grade subtraction should also use numbers such that the "upper" number is larger than the "lower" number.
- In a file whose name is `Log.txt`, append a line of text summarizing each user, include the user's name, grade-level, and score.
- Provide more helpful feedback to the user. For example, in the first example in the above interaction you might review the concept of "carrying".

When you write your program, be sure you:

- think about the functional-decomposition (top-down) design **before you start to write code!** You will need to turn in a design document (see below for details).
- use meaningful variable names with good style (i.e., useCamelCase)
- use comments (""" Multi-line Comment """) at the start of the program **and** immediately after each function definition describing what they do (see lab1 `diceOutcomes.py` program)
- use a main function (see lab1 `diceOutcomes.py` program) located at the top of program with a call to it at the bottom to start execution
- use global constants where appropriate with good style (ALL_CAPS_AND_UNDERSCORES). (Put your global constants after your initial comments describing the program and before your main function definition so they can be found and changed easily in future versions of your program.)

Submit the single file, **hw2.zip** containing the following:

- **mathTutor.py** (your Python program)
- **design.doc** (or `design.pdf`, or `design.txt`, or `design.rtf`) a document describing the design of your program including a **functional-decomposition diagram showing parameters and returned values**, and text describing each function (see lab1 description)

The steps for the homework submission system are:

1. Design, write, debug, and test your program in the hw2 folder. When you are ready to submit your homework, zip the whole folder by right-clicking on it and selecting `Send to | Compressed (zipped) folder`. This will create a new file called `hw2.zip` which you will submit electronically. (This assumes Windows OS....)
2. Log on to the submission system at: https://www.cs.uni.edu/~schafer/submit/which_course.cgi
(It is very likely that you will get some security certificate warnings when trying to use this. You may add an exception and accept the existing security certificate.) Use the same CatID user-name and password you use to log on the lab computers.
3. Select the course and section number of "CS 1520, Data Structures, Fienup". Click the "Continue".
4. Select the homework that you wish to submit: "HW 2: Math Tutor". Click the "Continue" button.
5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
6. Upload your program by Browsing and selecting your `hw2.zip` file. Click the "Continue" button.
7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.
(If you miss the deadline, you'll need to submit it as above, but select "Late Homeworks" in step 4 above.)