**1. Temperature conversion**

Redo exercise 3 of lab #4, with the following change: you must now define and use the function toFahrenheit in your program. The header of toFahrenheit is as follow:

double toFahrenheit(double x);

This function is used to convert a temperature in Celsius to Fahrenheit. toFahrenheit(x) is the temperature in Fahrenheit equivalent to x degrees Celsius. You must start from your version of exercise 3 from lab #4 and try to adapt it by introducing the new function toFahrenheit. The function should be defined in two steps: first the header before the main function, second the full function definition (header plus body) after the main function.

**2. Prime numbers**

Redo exercise 5 of lab #4, with the following change: you must now define and use the function isPrime in your program. The header of isPrime is as follow:

int isPrime(int n);

This function is used to check if the parameter n is prime. The function returns n if n is prime, p otherwise, such that p is not 1 nor n and p divides n. You must start from your version of exercise 5 from lab #4 and try to adapt it by introducing the new function isPrime. The function should be defined in two steps: first the header before the main function, second the full function definition (header plus body) after the main function.

**3. Applicant score**

You are to write a program that prompts the user for information about two applicants and that computes an overall score for each applicant. For each applicant, we prompt for exam scores (either SAT or ACT) and overall GPA. The exam information is turned into a number between 0 and 100 and the GPA information is turned into a number between 0 and 100 and these two scores are added together to get an overall score between 0 and 200. After obtaining scores for each applicant, the program reports which one looks better or whether they look equal. These scores are turned into a real-valued number between 0 and 100 using the following formulas:

* for SAT score :

SAT formula

* for ACT score :

ACT formula  
  
After computing this exam score, we compute a number between 0 and 100 based on the GPA. The program prompts for the GPA, the maximum GPA, and a transcript multiplier. All three of these values are real values (i.e., they can have a decimal part). The transcript multiplier is a value between 0.8 and 1.0 that the admissions staff use to account for differences across students and across schools. You should turn this into a score between 0 and 100 using the following formula:

* GPA result :

GPA formula  
  
At this point your program has two scores that vary from 0 to 100, one from test scores and one from GPA. The overall score for the applicant is computed as the sum of these two numbers (exam result + gpa result). Because each of these numbers is between 0 and 100, the overall score for an applicant ranges from 0 to 200. As indicated in the sample log of execution, your program is to report the exam and GPA subscores and the overall score for each applicant. These should be rounded to one decimal place. In addition to reporting the score for each applicant, the program should also produce whichever of the following messages is appropriate:

* *The first applicant seems to be better*
* *The second applicant seems to be better*
* *The two applicants seem to be equal*

Again, you MUST complete all the functions whose profile appears in the main file provided (see the supporting files below). You are not supposed to add more functions. Check the log of execution files below.

**Supporting files**

* [lab05-main-ex03.cpp](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-main-ex03.cpp)
* [lab05-ex03-log1.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex03-log1.txt)
* [lab05-ex03-log2.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex03-log2.txt)
* [lab05-ex03-log3.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex03-log3.txt)
* [lab05-ex03-log4.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex03-log4.txt)

**4. Guessing game**

Redo exercise 4 of lab #4, with the following changes:

* the user can play multiple games
* the user must play **at least** one game
* when the user stops playing, the program reports various statistics about the series of games played.

At the end of all games, you are to report the total number of games played, the total number of guesses made (all games included), the average number of guesses per game, and the best (fewest) number of guesses used in any single game. The average number of games should be displayed with one decimal.

Check the log file [lab05-ex04-log1.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex04-log1.txt)  and notice that the query to the user to know if she/he wants to play again is protected against bad input values: if the user doesn't input one of the letters y, Y, n or N, the program keep asking the question again and again.

You are to use your solution (or the online solution) of exercise 4 of lab #4 and add functions to structure your code.

**Supporting files**

* [lab05-ex04-log1.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex04-log1.txt)
* [lab05-ex04-log2.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex04-log2.txt)
* [lab05-ex04-log3.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex04-log3.txt)

**5. Math repetitor**

Redo exercise 1 of lab #4 with the following changes:

* now the user can choose the operation she/he wants to train with
* the program will check the input choice entered by the user and ask for another input if it’s wrong

Here is a log of execution of this program (user input in bold):

Welcome to Math Tutor  
------------------------------  
1. Addition  2. Subtraction  3. Multiplication  4. Division  0. Quit this program  
------------------------------  
Enter your choice (0-4): **1**  
  
  329  
+  43  
 ----  
  **472**  
  
Sorry, the correct answer is 372.  
  
------------------------------  
1. Addition  2. Subtraction  3. Multiplication  4. Division  0. Quit this program  
------------------------------  
Enter your choice (0-4): **3**  
  
   82  
x   2  
  ----  
  **164**  
  
Congratulations! That’s right.  
  
------------------------------  
1. Addition  2. Subtraction  3. Multiplication  4. Division  0. Quit this program  
------------------------------  
Enter your choice (0-4): **5**  
Enter your choice (0-4): **-1**  
Enter your choice (0-4): **0**  
  
Thank you for using Math Tutor.

You must start from your version of exercise 1 from lab #4 and try to adapt it. There is no function headers provided for this exercise and you must introduce yourself functions to solve nicely the problem. Check the log of execution files below.

**Supporting files**

* [lab05-ex05-log1.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex05-log1.txt)
* [lab05-ex05-log2.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex05-log2.txt)
* [lab05-ex05-log3.txt](http://users.polytech.unice.fr/~gaetano/db/cse142/labs/lab05/lab05-ex05-log3.txt)

**6. Rocket Ship**

You are to write a program that display a rocket ship on the screen, like this one:

     /\*\*\

    //\*\*\\

   ///\*\*\\\

  ////\*\*\\\\

 /////\*\*\\\\\

+=\*=\*=\*=\*=\*=\*+

|../\..../\..|

|./\/\../\/\.|

|/\/\/\/\/\/\|

|\/\/\/\/\/\/|

|.\/\/..\/\/.|

|..\/....\/..|

+=\*=\*=\*=\*=\*=\*+

|\/\/\/\/\/\/|

|.\/\/..\/\/.|

|..\/....\/..|

|../\..../\..|

|./\/\../\/\.|

|/\/\/\/\/\/\|

+=\*=\*=\*=\*=\*=\*+

     /\*\*\

    //\*\*\\

   ///\*\*\\\

  ////\*\*\\\\

 /////\*\*\\\\\

You are to exactly reproduce this output.  The various sub-figures in this output have a size of 3.  These sub-figures have the property that their size determines a parameter we call the *size* of the rocket, so there is one size variable.  All the function you write must have that size as parameter. The previous figure is a rocket of size 3 and can be obtained by providing the value 3 as the size requested by the program:

Welcome to the Rocket Ship Program

Enter the size of the rocket: **3**

     /\*\*\

    //\*\*\\

   ///\*\*\\\

  ////\*\*\\\\

 /////\*\*\\\\\

+=\*=\*=\*=\*=\*=\*+

|../\..../\..|

|./\/\../\/\.|

|/\/\/\/\/\/\|

|\/\/\/\/\/\/|

|.\/\/..\/\/.|

|..\/....\/..|

+=\*=\*=\*=\*=\*=\*+

|\/\/\/\/\/\/|

|.\/\/..\/\/.|

|..\/....\/..|

|../\..../\..|

|./\/\../\/\.|

|/\/\/\/\/\/\|

+=\*=\*=\*=\*=\*=\*+

     /\*\*\

    //\*\*\\

   ///\*\*\\\

  ////\*\*\\\\

 /////\*\*\\\\\

On any given execution your program will produce just one version of this figure, but it is possible to change the size of the rocket to have your program produce a figure of a different size.  For example, with size equal to 5 the output should look like this:

Welcome to the Rocket Ship Program

Enter the size of the rocket: **5**

         /\*\*\

        //\*\*\\

       ///\*\*\\\

      ////\*\*\\\\

     /////\*\*\\\\\

    //////\*\*\\\\\\

   ///////\*\*\\\\\\\

  ////////\*\*\\\\\\\\

 /////////\*\*\\\\\\\\\

+=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*+

|..../\......../\....|

|.../\/\....../\/\...|

|../\/\/\..../\/\/\..|

|./\/\/\/\../\/\/\/\.|

|/\/\/\/\/\/\/\/\/\/\|

|\/\/\/\/\/\/\/\/\/\/|

|.\/\/\/\/..\/\/\/\/.|

|..\/\/\/....\/\/\/..|

|...\/\/......\/\/...|

|....\/........\/....|

+=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*+

|\/\/\/\/\/\/\/\/\/\/|

|.\/\/\/\/..\/\/\/\/.|

|..\/\/\/....\/\/\/..|

|...\/\/......\/\/...|

|....\/........\/....|

|..../\......../\....|

|.../\/\....../\/\...|

|../\/\/\..../\/\/\..|

|./\/\/\/\../\/\/\/\.|

|/\/\/\/\/\/\/\/\/\/\|

+=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*+

         /\*\*\

        //\*\*\\

       ///\*\*\\\

      ////\*\*\\\\

     /////\*\*\\\\\

    //////\*\*\\\\\\

   ///////\*\*\\\\\\\

  ////////\*\*\\\\\\\\

 /////////\*\*\\\\\\\\\

Your program should have many functions apart from the main function, and the top function to draw the rocket should have the following profile:

void draw\_rocket(int size);