University of Pennsylvania
EAS 105: Fall 2010
MATLAB PROJECT: Integer Base Conversion
Instructor: Donna Dietz

Before you begin this project, you should become familiar with converting integers between different base systems. You should find all relevant information in your class notes, but additionally, there is plenty of information on the web. For this project, all integers will be represented as horizontal vectors. So, for example, in base 10, the number 345 would be represented by the vector $[3, 4, 5]$. The base 16 number which is normally shown as $23A0BF$ would be represented as $[2, 3, 10, 0, 11, 15]$. In all cases, the far right portion of the vector, as displayed on the screen (with the highest index) contains the least significant data.

You should need only six functions for this project. When your project passes the Tester, you should zip it all up and submit it through Blackboard, under *Course Documents*. If you submit it after the due date but less than a week late, you will be penalized 10% of the value. After a week, it will be penalized 20%. No projects may be submitted for credit after the final exam has been given.

For this project, create these six functions:

function [v_out] = ConvertVectorBaseTenToBase(v_in, base_in) For this function, you get to figure out how to convert a vector from base ten to any other integer base (2 or greater).

function [v_out] = ConvertVectorBaseToBaseTen(v_in, base_in) For this function, you get to figure out how to convert a vector from any integer base to base ten.

function [v_out] = ConvertVectorBaseAToBaseB(v_in, A, B) Although you are welcome to work this out the more 'proper' way, what I expect here is that you will simply make function calls to the two functions you have just created in order to make conversions between bases in which neither one of them is base ten.

function [v_out] = QuickLengthenVectorBaseAToBaseB(v_in, A, B) As was/will-be discussed in class, when the two bases are in a special relationship ($A^n = B$ or $B^n = A$), conversion can be done more quickly. For this function presume that $B^n = A$, for example, $A = 16$ and $B = 2$. Your function must find n. You may presume that this routine will only be handed appropriate input.

function [v_out] = QuickShortenVectorBaseAToBaseB(v_in, A, B) This function is the exact opposite of the one above. For example, $A = 3$ and $B = 81$.

function [v_out] = QuickConvertVectorBaseAToBaseB(v_in, A, B) This function simply tests for whether A or B is larger and passes the control on to one of the previous two functions.

With this entire project, you may disregard input error checking. In other words, your code will not be tested against input such as
$QuickLengthenVectorBaseAToBaseB([1, 2, 3, 4], 3, 6)$ or any other input which would cause a runtime error. Just have fun solving the problem as it stands and coding it up.

The *Tester* function for this code not only tests for correctness of your answers, but it also tests that your *Quick* functions are actually faster than your other ones. So, please note, do not pre-test your input in the non-Quick cases and divert appropriate inputs to the *Quick* routines. Also (this should go without saying) when your submission is graded officially, it may be manually inspected to ensure that nothing 'odd' is going on. For example, do not add a loop simply to increase the time spent on the non-Quick routines so the test will pass. Since I'm actually giving you the *Tester* code, Anything of this sort would cause you to lose significant credit, even if the tests all pass. The *Tester* file is there to help guide your progress when there is nobody around (at midnight or 3am) to give you advice. Use it as it is intended.

HAVE FUN!!!