Assignment 5 clarifications

- You must test the validity of user input.
  If the user enters anything other than a numeric character (or initial sign), output a message & request correct input: loop until valid input is received.
  Unlike earlier assignments, do not just quit the program, or start over from scratch - just insist on getting a valid input for that digit.

- You are required to use the more efficient order for multiplication *(this is not a suggestion!)*.

- This is the big one:
  All "modules" of the program **must be written as subroutines:**

  - obtain input (as sign/numeric digits) from user, & convert to binary
    (validity test may be a separate subroutine, or not - as you please;
    *also, the conversion to binary can be performed "on the fly" as part of the input subroutine**; or** you can have one subroutine take the input from user and store it as a character array, and then build a separate subroutine for the conversion).*
  - test relative size of two numbers, swap if necessary *(what is this for??)*
  - multiply two numbers
  - convert 16-bit number to character array for output
    suggestion: make it a null-terminated array, so you can output it simply with TRAP x22, aka PUTS

  For each subroutine, you will have to plan out carefully:

- what arguments do you need to pass in?
- some arguments will be arrays - but you cannot actually "pass" an entire array - so what will you pass?

- what value(s) do you need to return?
- as for the arguments, some return values will be arrays - how will you return them?
- remember that some subroutines will require both a value and a flag indicating whether the result was valid

- remember to practice "register hygiene" - save & restore any registers you will be using in the subroutine - and ONLY those registers!

  Make sure you spend enough time planning your subroutines with pencil and paper before you touch your keyboard!!!

  Your goal should be to reduce your "main" program to little more than than a set of calls to your subroutines - probably no more than a dozen lines of code.