Problem: Given an integer value, the current number system of the integer (2-10), and the desired number system to which the value is to be converted (2-10), display the following:

1. The original number as input by the user along with its current number system (also known as its base).
2. The input number converted to the decimal number system (base 10).
3. The base 10 value from #2 after the ROT-5 algorithm has been applied to it (see below).
4. The base 10 value generated from #3 then converted to the desired number system as input by the user.

The ROT-5 algorithm does the following to a decimal number system (base 10) integer:

- Digits from 0 to 4 will each be added with 5.
- Digits from 5 to 9 will have 5 subtracted from each.
- Example: The number 9036 would be 4581 after the ROT-5 algorithm is applied.

Example Execution #1:

Enter a number: 674
Enter the base of the number: 10
Enter the desired base: 2

Original Number (Base-10): 674
Original Base-10:           674
Rotated Base-10:            129
Converted Base- 2:          10000001

Example #1 Explained:
The first two lines of generated output are the same due to the fact that the input was already in base 10.
The value 129 is the result of the ROT-5 algorithm applied to 674.
The final value 10000001 is the base 2 conversion of the 129 base 10 value.

Example Execution #2:

Enter a number: 10000001
Enter the base of the number: 2
Enter the desired base: 10

Original Number (Base- 2):       10000001
Original Base-10:                     129
Rotated Base-10:                      674
Converted Base-10:                    674

Example #2 Explained:
In this example the original number was input as base 2, converted to base 10 (129), and then has the ROT-5 algorithm applied to it.
The value after the ROT-5 has been applied is already in the desired base 10 and no further conversion is needed.

Example Execution #3:

Enter a number: 1001
Enter the base of the number: 8
Enter the desired base: 2

Original Number (Base- 8):           1001
Original Base-10:                     513
Rotated Base-10:                      68
Converted Base- 2:                1000100

Example #3 Explained:
Notice here that a digit is lost when the base 10 number 513 has the ROT-5 applied to it (68). There is no expectation that your program handle leading zero digits on this assignment.

More information on number systems can be found in Appendix D of your C programming text.
**Example Execution #4:**

Enter a number: 125  
Enter the base of the number: 8  
Enter the desired base: 10  

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Original Number (Base- 8):            125  
Original Base-10:                     85  
Rotated Base-10:                       30  
Converted Base-10:                     30  
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

**Example #4 Explained:**

- Begin with input of 125 in base 8 (octal).
- Convert from base 8 to base 10 (result is 85).
- Apply ROT-5 to generate 30.
- No additional work needed as desired base is also base 10.

---

**Example Execution #5:**

Enter a number: 375  
Enter the base of the number: 8  
Enter the desired base: 5  

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Original Number (Base- 8):            375  
Original Base-10:                     253  
Rotated Base-10:                      708  
Converted Base- 5:                    10313  
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

**Example #5 Explained:**

- Begin with input of 375 in base 8.
- Convert from base 8 to base 10 (result in 253).
- Apply ROT-5 to generate 708.
- Convert rotated base 10 value to desired ending base 5 value (10313).

Additional Requirements:

1. Add the homework assignment header file to the top of your program. A description of your program will need to be included in the assignment header. This particular header can be added to your file by entering `hhw` while in command mode in `vi`.

2. The example executions provided each represent a single test of your program. We will test your final submission with the data used in the test case above and with several unpublished test cases.
   - Your program is expected to accept input and produce output in the exact same manner demonstrated above.
   - You may assume that the user of your program will only enter valid integers for the base (2-10) and the corresponding value will be a number that can be represented in the specified base.
   - Do not add any “bonus” features not demonstrated in the example executions provided.

3. Course standards prohibit the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the first six chapters of the book, notes, and lectures to be acceptable for use.

4. For this assignment you will be required to implement the user-defined functions (from chapter 4). Failing to follow course standards as they relate to good user-defined function use will result in a zero for this assignment.

   With our ability to implement user-defined functions only the following will be permitted in the main function:
   
   1. Declaration of variables to be passed to functions.
   2. Calls to user-defined functions by main.
   3. A limited amount of control structures (see chapters 5 and 6) to retain the previous two tasks within the main function.

   Additionally, each user-defined function may represent a single task in your larger program. The failure to make a good use of user-defined functions as described here and in the course standards will result in a loss of ALL points on those assignments that require user-defined functions.

5. A program MUST compile, be submitted through the guru server as demonstrated during the first week of the semester in lab, and submitted prior to the posted due date to be considered for partial credit. **It cannot be said enough, start your work early!**
Course Programming and Documentation Standards Reminders:

- Use the course function header (head_fx vi shortcut hfx while in command mode) for every user-defined function in your program.
  - List and comment all parameters to a function, one per line, in the course function header.
  - All function declarations will appear in the global declaration section of your program.
  - The user-defined function definitions will appear in your program after the main function.
- Place a single space between all (binary) operators and operands.
- Comment all variables to the right of each declaration.
- Declare only one variable per line.

- Notice that several programs (see program 2-9 on pages 74-75) in the programming text use a single line comment to indicate the start of the local declaration and executable statement sections of the main function.
  - At no point during the semester should these two sections ever overlap. You might consider adopting this habit of commenting the start of each section to help you avoid this mistake.

- Select meaningful identifiers (names) for all variables in your program.
- Indent all code found within the main function exactly two spaces.
- Indent all code found within the body of relevant selection and repetition constructs exactly two additional spaces.
- Make use of { and } with all relevant section and repetition constructs.
- Do not single (or double) space the entire program, use blank lines when appropriate.
- There is no need to include example output with your submission.
- Remove any diagnostic print statements from your code, even if they are commented out (inactive), unless you believe your program to be logically incorrect or incomplete as a way to demonstrate to your lab instructor how much of the problem you were able to solve and attempted to implement.

When you submit... only the final attempt of a submission is kept for grading. All other submissions are over-written and cannot be recovered. You may make multiple submissions but only the last attempt is retained and graded.

- Verify in the confirmation e-mail sent to you by the course that you have submitted the correct file (must be named hw05.c), to the correct assignment (hw05), and to the correct lab section (CRN value).
- Leave time prior to the due date to seek assistance should you experience difficulties completing or submitting this assignment.
- All attempts to submit via a method other than through the guru server as set up during the first lab of the semester will be denied consideration.

Assignment deadlines... are firm and the electronic submission will disable promptly as advertised. We can only grade what you submit as expected prior to the assignment deadline.

Academic Integrity Reminder:

- Please review the policies of the course as they relate to academic integrity. The assignment you submit should be your own original work. You are to be consulting only course staff regarding your specific algorithm for assistance. Collaboration is not permitted on individual homework assignments.

All course programming and documentation standards are in effect for this and each assignment this semester. Please review this document!