1. Write a program to input an uppercase letter and display its corresponding lowercase letter.
2. Write a program to read two decimal digits whose sum is less than 10 and display them and their sum in the next line with appropriate message.
3. Write a program that will prompt the user to enter a hex digit character (0-F), display it on the next line in decimal and ask the user if he or she wants to do it again. If the user types “Y”, the program repeats. If the user types “N” the program terminates. If the user enters an illegal character prompt the user to try again.
4. Write a program to read two capital letters and display them on the next line in alphabetical order.
5. Write a program that prompts the user to enter a character, and in subsequent lines prints its ASCII code in binary and the number of 1 bits in its ASCII code.
6. Write a program that prompts the user to type a hex number of four hex digits or less, and outputs it in binary on the next line. If the user enters an illegal character, he or she should be prompted to begin again. Accept only uppercase letters. Your program may ignore any input beyond four characters.
7. Write a program that prompts the user to enter two unsigned hex numbers, 0 to FFFFh, and prints their sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again. Your program should be able to handle the possibility of unsigned overflow. Each input ends with a carriage return.
8. Write a program that prompts the user to enter a string of decimal digits, ending with a carriage return, and prints their sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again.
9. Write a program that lets the user type some text, consisting of words separated by blanks, ending with a carriage return, and displays the text in the same word order as entered, but with the letters in each word reversed.
10. Write a program that lets the user type in an algebraic expression, ending with a carriage return that contains round (parentheses) square, and curly brackets. As the expression is being typed in, the program evaluates each character. If at any point the expression is incorrectly bracketed (too many right brackets or a mismatch between left and right brackets), the program tells the user to start over. After the carriage return is typed, if the expression is correct, the program displays “expression is correct." If not, the program displays “too many left brackets". In both cases, the program asks the user if he or she wants to continue. If the user types ‘Y’, the program runs again. Your program does not need to store the input string, only check it for correctness.
11. Write a program that lets the user enter time in seconds, up to 65535 and outputs the time as hours, minutes, and seconds.
12. Write a program to find the greatest common divisor (GCD) of two integers M and N, according to the following algorithm:
    1. Divide M by N, getting quotient Q and remainder R.
    2. If R = 0 then stop. N is the GCD of M and N.
    3. If R <> 0 replace M by N, N by R, and repeat step 1
13. Suppose the class records are stored as follows

CLASS

DB ‘MARY ALLEN’, 67,45,98,33

DB 'SCOTT BAYLIS’, 70,56,87,44

DB 'GEORGE FRANK', 82,72,89,40

DB 'SAM WONG', 78,76,92,60

Each name occupies 12 bytes. Write a program to print the name of each student and his or her average (truncated to an integer) for the four exams.

1. Write a program that (a) lets the user input a string, (b) prints it forward and backward without punctuation and blanks on successive lines, and (c) decides whether it is a palindrome and prints the conclusion.