This program uses the two subroutines written for program 3 in a simple encryption application.

Your program should

1. Set up a plain text string P of about 30 characters (use .ASCII). To make the following processing easier, your string should have an even number of characters and be terminated by TWO null bytes.

2. Output P using stro and output the words of P in hexadecimal using the HEXO subroutine from Program 3.

3. Input a decimal encryption key K (integer).

4. Create an encrypted version of P by exclusive-or-ing each word of P with K. Use EXOR from program 3. Store the resultant string followed by two null bytes in character array C.

5. Output the words of C in hexadecimal – see #2.

6. Create a third array of characters (P2) by exclusive-or-ing each word of C with K. Add two null bytes to P2. P2 should be the same length as P.

7. Output P2 using stro.

Your program should use subroutines where appropriate. For example, steps 2 and 5 are similar; steps 4 and 6 are similar.

Example program run.

This is the test string for program four
546869732069732074686520737420737472696e6720666f7220726f6772616d20666f7573
Enter encryption key: 32767
2b97168c5f960cdf0b971adf0b9a0c8b5f8c0b8d169118df19900ddf0f8d10980d9e12df19900a8d
This is the test string for program four

By the due date

Turn in a listing of your program and the results of testing it

Grading

Correctness: 50 points
Testing: 15 points
Readability: 35 points – much of the points are for correct use of subroutines.