```c
#include<stdio.h>

int main()
{
    int data[7];
    int dataatrec[7], c, c1, c2, c3, i;
    printf("\n********** HAMMING CODE ***********");

    // Input 4 data bits
    printf("\n\nEnter 4 Bits Of Data One By One\n");
    scanf("%d", &data[0]);
    scanf("%d", &data[1]);
    scanf("%d", &data[2]);
    scanf("%d", &data[4]);

    // Calculate even parity bits
    data[6] = data[0] ^ data[2] ^ data[4];  // P1
    data[5] = data[0] ^ data[1] ^ data[4];  // P2
    data[3] = data[0] ^ data[1] ^ data[2];  // P4

    // Output the encoded data
    printf("\nEncoded Data is: ");
    for(i = 0; i < 7; i++) {
        printf("%d", data[i]);
    }

    // Input the received data
    printf("\n\nEnter Received Data Bits One By One\n");
    for(i = 0; i < 7; i++) {
        scanf("%d", &dataatrec[i]);
    }

    // Calculate the syndrome bits
    c1 = dataatrec[6] ^ dataatrec[4] ^ dataatrec[2] ^ dataatrec[0];
    c2 = dataatrec[5] ^ dataatrec[4] ^ dataatrec[1] ^ dataatrec[0];
    c3 = dataatrec[3] ^ dataatrec[2] ^ dataatrec[1] ^ dataatrec[0];

    c = c3 * 4 + c2 * 2 + c1;  // Calculate the position of the error

    if (c == 0) {
        printf("\n\nNo Error While Transmitting Data\n");
    } else {
        printf("\nError on position %d", c);
        printf("\nData sent :");
        for(i = 0; i < 7; i++) {
            printf("%d", data[i]);
        }
        printf("\nData received :");
        for(i = 0; i < 7; i++) {
            printf("%d", dataatrec[i]);
        }

        // Correct the error
        printf("\nCorrected Message is : ");
        dataatrec[7 - c] = dataatrec[7 - c] == 0 ? 1 : 0; // Flip the erroneous bit

        for(i = 0; i < 7; i++) {
            printf("%d", dataatrec[i]);
        }
    }

    return 0;
}
```