```c
/**********************************************************
 *                                                        *
 * HW04 Q2                                                *
 * Student Name: MUSTAFA AkILLI                           *
 * Student ID  : 131044017                                *
 * Date        : 15 March 2015                            *
 * Points      : 19                                       *
 *                                                        *
 **********************************************************/
#include <stdio.h>

#define PLAINTEXTFILE "Files/Q2/ReceivedMessage.txt"
#define ENCODEDFILE "Files/Q2/EncodedInput.txt"
#define CRYPTEDINPUT "Files/Q2/CryptedInput.txt"

void decode_and_write_to_file(FILE *f_out_ptr, int number_of_ones);
int decode_message(FILE *f_in_ptr, FILE *f_out_ptr);
int decrypt_message(FILE *f_in_ptr, FILE *f_out_ptr);



int
main(int argc, char* argv[])
{
    FILE *f_plane_ptr, *f_encoded_ptr, *f_crypted_ptr;

    f_crypted_ptr = fopen(CRYPTEDINPUT,"r");

    if(f_crypted_ptr == NULL){

        printf("ERROR!! Plain text file could not be opened to read.\n");
        return 0;
    }


    f_encoded_ptr = fopen(ENCODEDFILE,"w");

    if(f_encoded_ptr == NULL){

        printf("ERROR!! Encoded text file could not be opened to write.\n");
        return 0;
    }


    decrypt_message(f_crypted_ptr, f_encoded_ptr);

    fclose(f_crypted_ptr);
    fclose(f_encoded_ptr);


    f_encoded_ptr = fopen(ENCODEDFILE,"r");

    if(f_encoded_ptr == NULL){

        printf("ERROR!! Plain text file could not be opened to read.\n");
        return 0;
    }

    f_plane_ptr = fopen(PLAINTEXTFILE,"w");

    if(f_plane_ptr == NULL){

        printf("ERROR!! Encoded text file could not be opened to write.\n");
        return 0;
    }

    decode_message(f_encoded_ptr, f_plane_ptr);

    fclose(f_encoded_ptr);
    fclose(f_plane_ptr);

    return 0;
```

```c
}

/**********************************************************
 * Gets FILE* to write file and character to decode          *
 * uses encoding table to convert encoded message to         *
 * plain text message                                        *
 **********************************************************/

void decode_and_write_to_file(FILE *f_out_ptr, int number_of_ones)
{

    switch(number_of_ones){

        case 1 :  fprintf(f_out_ptr,"E");
                  break;
        case 2 :  fprintf(f_out_ptr,"I");
                  break;
        case 3 :  fprintf(f_out_ptr," ");
                  break;
        case 4 :  fprintf(f_out_ptr,"T");
                  break;
        case 5 :  fprintf(f_out_ptr,"C");
                  break;
        case 6 :  fprintf(f_out_ptr,"N");
                  break;
        case 7 :  fprintf(f_out_ptr,"A");
                  break;
        case 8 :  fprintf(f_out_ptr,"G");
                  break;
        case 9 :  fprintf(f_out_ptr,"B");
                  break;
        case 10 : fprintf(f_out_ptr,"Z");
                  break;
        case 11 : fprintf(f_out_ptr,"H");
                  break;
        case 12 : fprintf(f_out_ptr,"L");
                  break;
        case 13 : fprintf(f_out_ptr,"U");
                  break;
        case 14 : fprintf(f_out_ptr,"V");
                  break;
        case 15 : fprintf(f_out_ptr,"R");
                  break;
        case 16 : fprintf(f_out_ptr,"S");
                  break;
        case 17 : fprintf(f_out_ptr,"Y");
                  break;
    }
}

/**********************************************************
 * Gets FILE* f_in_ptr to read from encoded text file and    *
 * FILE* f_out_ptr to write message to plain text file       *
 * return number of characters read from encoded text        *
 **********************************************************/
int decode_message(FILE *f_in_ptr, FILE *f_out_ptr)
{

    int counter = 0;
    int Encoded;
    int status;
    int number_of_ones=1;

    status = fscanf(f_in_ptr,"%1d",&Encoded);

    while(status != EOF){

        status = fscanf(f_in_ptr,"%1d",&Encoded);
        ++number_of_ones;

        if(Encoded==0){
            if(status != EOF){
                decode_and_write_to_file(f_out_ptr,number_of_ones);
                number_of_ones=0;
            }
        }
    }
    return counter;
}

/*************************************************************
 * Gets FILE* f_in_ptr to read from encrypted text file and  *
 * FILE* f_out_ptr to write message to encoded file          *
 * return encrypted character number                         *
 *************************************************************/
int decrypt_message(FILE *f_in_ptr, FILE *f_out_ptr)
{

    int counter = 0;
    char Crypted;
    char status;
    int M=0;
    int N=5;
    int temp=5;


    status = fscanf(f_in_ptr,"%c",&Crypted);

    while(status != EOF){

        if(Crypted=='*'){
            fprintf(f_out_ptr,"1");
        }

        else if(Crypted=='_'){
            fprintf(f_out_ptr,"0");
        }

        --N;

        if(N==M){

            status = fscanf(f_in_ptr,"%c",&Crypted);
            --temp;

            if(temp>N){
                N = temp;
            }
            else{
                temp=5;
                N=5;
            }
        }

        status = fscanf(f_in_ptr,"%c",&Crypted);
        ++counter;
    }

    return counter;
}
```