```
/**************************************************************
 *                                                            *
 * HW04 Q3                                                    *
 * Student Name: MUSTAFA AKILLI                               *
 * Student ID  : 131044017                                    *
 * Date        : 15 MARCH 2015                                *
 * Points      : 49                                           *
 *                                                            *
 **************************************************************/
#include <stdio.h>

#define TRUE 1
#define FALSE 0
#define CHARACTERFILE "Files/Q3/CharacterList.txt"
#define SAMPLEFILE "Files/Q3/Sample.txt"
#define ENCODEDFILE "Files/Q3/XUniversityEncoded.txt"
#define PLAINTEXTFILE "Files/Q3/XUniversityMessage.txt"

void swap_int(int *a, int *b);
void swap_char(char *a, char *b);
void sort(char *a, int a_num, char *b, int b_num, char *c, int c_num);
int is_letter(char c);
int read_character_list(FILE* f_in_ptr, char *c1, char *c2, char *c3);
void count_letters(FILE *f_in_ptr, char *c1, char *c2, char *c3);
void decode(FILE *f_in_ptr, FILE *f_out_ptr, char c1, char c2, char c3);


int
main(int argc, char* argv[])
{
    FILE *f_character_list_ptr, *f_sample_file_ptr, *f_encoded_ptr,
        *f_plain_text_ptr;
    int character_number;
    char c1, c2, c3;

    f_character_list_ptr = fopen(CHARACTERFILE,"r");

    if(f_character_list_ptr == NULL){

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

    character_number = read_character_list(f_character_list_ptr,&c1,&c2,&c3);

    if(character_number != 3){

        printf("ERROR!! Number of letter read is not equal to three.\n");
        return 0;
    }

    fclose(f_character_list_ptr);

    f_sample_file_ptr = fopen(SAMPLEFILE,"r");

    if(f_character_list_ptr == NULL){

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

    count_letters(f_sample_file_ptr,&c1,&c2,&c3);

    fclose(f_sample_file_ptr);

    f_encoded_ptr = fopen(ENCODEDFILE,"r");
    f_plain_text_ptr = fopen(PLAINTEXTFILE,"w");

    if(f_encoded_ptr == NULL){

        printf("ERROR!! Encoded file could not be opened to read.\n");
        return 0;
```

```
    }

    if(f_plain_text_ptr == NULL){

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

    decode(f_encoded_ptr,f_plain_text_ptr,c1,c2,c3);

    fclose(f_encoded_ptr);
    fclose(f_plain_text_ptr);

    return 0;
}

/*************************************************************
 * Swaps values of two integers                              *
 *************************************************************/
void swap_int(int *a, int *b)
{
    int t;
    t  = *a;
    *a = *b;
    *b = t;
}

/*************************************************************
 * Swaps values of two characters                            *
 *************************************************************/
void swap_char(char *a, char *b)
{
    int t;
    t  = *a;
    *a = *b;
    *b = t;
}

/*************************************************************
 * Sorts characters according to counts. At the end          *
 * make sure that *c1 keeps most frequent used letter, *c3   *
 * keeps least frequent used letter and *c2 keeps remained   *
 * letter                                                    *
 *************************************************************/
void sort(char *a, int a_num, char *b, int b_num, char *c, int c_num)
{
    if(b_num>a_num){
        swap_char(a,b);
    }

    if(c_num>a_num){
        swap_char(a,c);
    }

    if(c_num>b_num){
        swap_char(b,c);
    }
}

/*************************************************************
 * Check whether character is big ASCII letter or not        *
 * return TRUE or FALSE                                      *
 *************************************************************/
int is_letter(char c)
{
    if(c>='A' && c<='Z'){
        return TRUE;
    }

    else{
        return FALSE;
    }
```

```
}

/************************************************************
 * Read characters from character list file and if character *
 * is letter assign characters to c1, c2 and c3.             *
 * If file has not three letters assign NULL to input char   *
 * by order. For ex. file has two letters assign proper      *
 * letters to c1 and c2 and assign NULL to c3. If file has   *
 * four letters assign c1, c2 and c3 first three letters.    *
 * Return number of letters in character list file.          *
 * Do not forget to count only proper letters with your      *
 * is_letter function. Return number of letters not chars    *
 ************************************************************/
int read_character_list(FILE* f_in_ptr, char *c1, char *c2, char *c3)
{
    int counter = 0;
    char status;
    int letter;

    do{
        status = fscanf(f_in_ptr,"%c",c1);

        if(status == EOF){

            *c1 = ' ';
            return counter;
        }

        letter = is_letter(*c1);
    }while(letter==FALSE);


    ++counter;

    do{
        status = fscanf(f_in_ptr,"%c",c2);

        if(status == EOF){

            *c2 = ' ';
            return counter;
        }

        letter = is_letter(*c2);
    }while(letter==FALSE);


    ++counter;

    do{
        status = fscanf(f_in_ptr,"%c",c3);

        if(status == EOF){

            *c3 = ' ';
            return counter;
        }

        letter = is_letter(*c3);
    }while(letter==FALSE);

    ++counter;
    return counter;
}

/************************************************************
 * Read letters from Sample file and compute frequency of    *
 * letters. Then sort it inside this function. Call sort     *
 * function. At the end make sure that *c1 keeps most        *
 * frequent used letter, *c3 keeps least frequent used       *
 * letter and *c2 keeps remained letter                      *
 ************************************************************/
```

```c
void count_letters(FILE *f_in_ptr, char *c1, char *c2, char *c3)
{
    char status,temp;
    int a_num=0,b_num=0,c_num=0;

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

    while(status != EOF){

        if(temp == *c1){
            ++a_num;
        }

        if(temp == *c2){
            ++b_num;
        }

        if(temp == *c3){
            ++c_num;
        }

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

    sort(c1,a_num,c2,b_num,c3,c_num);
}

/************************************************************
 * Read from XUniversityEncoded file to decode message and   *
 * write decoded (plain text) message to XUniversityMessage  *
 * file. Make sure c1 keeps most frequent used letter,  c3   *
 * keeps least frequent used letter and  c2 keeps remained   *
 * letter while calling function. According to frequency     *
 * you know their codes. c1: 0, c2: 10, c3: 110.             *
 ************************************************************/
void decode(FILE *f_in_ptr, FILE *f_out_ptr, char c1, char c2, char c3)
{
    int counter=0;
    int temp,status;

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

    while(status != EOF){

        ++counter;
        if(temp==0){
            switch(counter){

                case 1 : fprintf(f_out_ptr,"%c",c1);
                         break;

                case 2 : fprintf(f_out_ptr,"%c",c2);
                         break;

                case 3 : fprintf(f_out_ptr,"%c",c3);
                         break;
            }
            counter=0;
        }

        status = fscanf(f_in_ptr,"%1d",&temp);
    }
}
```