#include <stdio.h>

#include <stdlib.h>

#include <math.h>

/\*-------STRUCTURES---------\*/

typedef struct {int rows; int cols; unsigned char\* data;} sImage;

/\*-------PROTOTYPES---------\*/

long getImageInfo(FILE\*, long, int);

void copyImageInfo(FILE\* inputFile, FILE\* outputFile);

void copyColorTable(FILE\* inputFile, FILE\* outputFile, int nColors);

int main(int argc, char\* argv[])

{

FILE \*bmpInput, \*bmpOutput;

sImage originalImage;

unsigned char someChar;

unsigned char\* pChar;

int nColors; /\* BMP number of colors \*/

long fileSize; /\* BMP file size \*/

int vectorSize; /\* BMP vector size \*/

int r, c; /\* r = rows, c = cols \*/

/\* initialize pointer \*/

someChar = '0';

pChar = &someChar;

if(argc < 2)

{

printf("Usage: %s bmpInput.bmp\n", argv[0]);

exit(0);

}

printf("Reading filename %s\n", argv[1]);

/\*--------READ INPUT FILE------------\*/

bmpInput = fopen(argv[1], "rb");

fseek(bmpInput, 0L, SEEK\_END);

/\*--------DECLARE OUTPUT FILE--------\*/

bmpOutput = fopen("output\_test.bmp", "wb");

/\*--------GET BMP DATA---------------\*/

originalImage.cols = (int)getImageInfo(bmpInput, 18, 4);

originalImage.rows = (int)getImageInfo(bmpInput, 22, 4);

fileSize = getImageInfo(bmpInput, 2, 4);

nColors = getImageInfo(bmpInput, 46, 4);

vectorSize = fileSize - (14 + 40 + 4\*nColors);

/\*-------PRINT DATA TO SCREEN-------------\*/

printf("Width: %d\n", originalImage.cols);

printf("Height: %d\n", originalImage.rows);

printf("File size: %ld\n", fileSize);

printf("# Colors: %d\n", nColors);

printf("Vector size: %d\n", vectorSize);

copyImageInfo(bmpInput, bmpOutput);

copyColorTable(bmpInput, bmpOutput, nColors);

/\*----START AT BEGINNING OF RASTER DATA-----\*/

fseek(bmpInput, (54 + 4\*nColors), SEEK\_SET);

/\*----------READ RASTER DATA----------\*/

for(r=0; r<=originalImage.rows - 1; r++)

{

for(c=0; c<=originalImage.cols - 1; c++)

{

/\*-----read data, reflect and write to output file----\*/

fread(pChar, sizeof(char), 1, bmpInput);

\*pChar = 255 - \*pChar;

fwrite(pChar, sizeof(char), 1, bmpOutput);

}

}

fclose(bmpInput);

fclose(bmpOutput);}

/\*----------GET IMAGE INFO SUBPROGRAM--------------\*/

long getImageInfo(FILE\* inputFile, long offset, int numberOfChars)

{

unsigned char \*ptrC;

long value = 0L;

unsigned char dummy;

int i;

dummy = '0';

ptrC = &dummy;

fseek(inputFile, offset, SEEK\_SET);

for(i=1; i<=numberOfChars; i++)

{

fread(ptrC, sizeof(char), 1, inputFile);

/\* calculate value based on adding bytes \*/

value = (long)(value + (\*ptrC)\*(pow(256, (i-1))));

}

return(value);

} /\* end of getImageInfo \*/

/\*-------------COPIES HEADER AND INFO HEADER----------------\*/

void copyImageInfo(FILE\* inputFile, FILE\* outputFile)

{

unsigned char \*ptrC;

unsigned char dummy;

int i;

dummy = '0';

ptrC = &dummy;

fseek(inputFile, 0L, SEEK\_SET);

fseek(outputFile, 0L, SEEK\_SET);

for(i=0; i<=50; i++)

{

fread(ptrC, sizeof(char), 1, inputFile);

fwrite(ptrC, sizeof(char), 1, outputFile);

}

}

/\*----------------COPIES COLOR TABLE-----------------------------\*/

void copyColorTable(FILE\* inputFile, FILE\* outputFile, int nColors)

{

unsigned char \*ptrC;

unsigned char dummy;

int i;

dummy = '0';

ptrC = &dummy;

fseek(inputFile, 54L, SEEK\_SET);

fseek(outputFile, 54L, SEEK\_SET);

for(i=0; i<=(4\*nColors); i++) /\* there are (4\*nColors) bytesin color table \*/

{

fread(ptrC, sizeof(char), 1, inputFile);

fwrite(ptrC, sizeof(char), 1, outputFile);

}

}

**OUTPUT:-**

