

Your program MUST open a **pgm** file called hw9.pgm and determine how many bytes it has.

Your program MUST then extract the hidden text by determining if the Least Significant Bit (LSB) of each set of 8 bytes of the image are 1s or a 0s (we are hiding 1 bit per byte of the image. So it takes 8 image bytes to extract 1 byte of hidden text). From that you can create a byte/character of text.

Your program MUST display each character of the hidden text. Note that there are 3 xs at the end of the hidden message. So, you can look for xxx and know when to stop extracting the hidden message.

You will need the following includes:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

You will want the following routines:

```

/*****
/* Clear PGM (XV) comments. */
*****/
void pgmCommentClear(FILE *disk){
    uchar ch;
    fread(&ch, 1, 1, disk);
    if (ch != '#') {
        fseek(disk, -1, SEEK_CUR);
        return;
    }
    do {
        while (ch != '\n') fread(&ch, 1, 1, disk);
    } while (ch == '#');
    pgmCommentClear(disk);
}

/*****
/* Read PGM formatted image (1D array). */
*****/
uchar *PGM_FILE_READ_1D(char *FileName, int *Width, int *Height, int *color)
{
    int pmax;
    char ch;
    char type[3];
    uchar *Image;
    FILE *disk;
    if ((disk = fopen(FileName, "rb")) == NULL) {
        return NULL;
    }
    fscanf(disk, "%s", type);
    if (!strcmp(type, "P6")) *color = 1;
        else *color = 0;
    fread(&ch, 1, 1, disk);
    pgmCommentClear(disk);
}

```

```

    fscanf(disk, "%d", Width);
    fscanf(disk, "%d", Height);
    fscanf(disk, "%d", &pmax);
    fread(&ch, 1, 1, disk);
    if (*color == 1) {
        Image = (uchar *)calloc(*Height * *Width * 3, sizeof(uchar));
        fread(Image, 1, (*Height * *Width * 3), disk);
    } else {
        Image = (uchar *)calloc(*Height * *Width, sizeof(uchar));
        fread(Image, 1, (*Height * *Width), disk);
    }
    fclose(disk);
    return Image;
}

/*****
/* Write PGM formatted image (1D array).*/
*****/
void PGM_FILE_WRITE_1D(char *FileName, uchar *Image, int Width, int Height,
int color) {
FILE *disk;
    disk = fopen(FileName, "wb");
    if (color == 1) fprintf(disk, "P6\n");
        else fprintf(disk, "P5\n");
    fprintf(disk, "%d %d\n", Width, Height);
    fprintf(disk, "255\n");
    if (color == 1) {
        fwrite(Image, 1, (Height * Width * 3), disk);
    } else {
        fwrite(Image, 1, (Height * Width), disk);
    }
    fclose(disk);
}

```

You will also want a routine to convert a decimal value to a bit string and a routine to convert a bit string to a decimal value.

EXTRA CREDIT (10 POINTS):

For extra credit add to your program the ability to hide your own message in a pgm file (such as the TajMahal.pgm image) by modifying the Least Significant Bit (LSB). For the extra credit you must modify the entire program such that it now requires a command-line argument and instructions as how to use the program. Such as:

./hw9 <-- no command line argument supplied

```

/*****
/* HW9 usage: */
/* ./hw9 normal <-- Normal operation for hw9 */
/* ./hw9 extra <-- Extra credit operation */
*****/

```

The file containing the text to hide MUST be named hw9.text
The resultant image must be named hw9.pgm

REQUIREMENTS:

1. Your program must run on the Linux Mint machines in Engineering.
2. Your full name must appear as a comment at the beginning of your program.
3. Your source code must be named hw9-yourname.c
4. Your source MUST compile using "gcc hw9-yourname.c"
5. Failure to follow the above directions will result in loss of points.