

```

#include <dos.h>
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <windows.h>

char far *Screen;

/* Sets the display to VGA 320x200x256 */
void VGAScreen(void)
{
    union REGS r;

    r.h.ah = 0;
    r.h.al = 0x13;
    int86(0x10, &r, &r);
    return;
}

/* Goes back to textmode */
void TextScreen(void)
{
    union REGS r;

    r.h.ah = 0;
    r.h.al = 0x3;
    int86(0x10, &r, &r);

    return;
}

/* This sets a DAC register to a specific Red Green Blue-value */
int SetDAC(unsigned char DAC, unsigned char R, unsigned char G, unsigned char B)
{
    outportb(0x3C8, DAC);
    outportb(0x3C9, R);
    outportb(0x3C9, G);
    outportb(0x3C9, B);

    return(0);
}

int LoadBMP(void)
{
    struct BMPHeader
    {
        unsigned short  bfType;
        long             bfSize, bfReserved, bfOffBits, biSize, biWidth, biHeight;
        unsigned short  biPlanes, biBitCount;
        long             biCompression, biSizeImage, biXPelsPerMeter,
                       biYPelsPerMeter, biClrUsed, biClrImportant;
    } Header;

    FILE *BMPFile;
    unsigned char c, Palette[256][4];
    char          *filename = "rck.bmp";
    unsigned int  offset, lines, paddedWidth;

    /* This checks for the file */
    BMPFile = fopen(filename, "rb");

    if (BMPFile == NULL)
    {
        printf("Cant open file.");
        return(1);
    }
}

```

```

/* Read the header information */
fread(&Header, 54, 1, BMPFile);

if (Header.bfType != 19778 || Header.bfReserved != 0 || Header.biPlanes != 1)
{
    /* Not a valid bitmap file - don't display */
    printf("Not a valid bitmap.");
    fclose(BMPFile);
    return(1);
}

if (Header.biCompression != 0)
{
    /*Compressed file - don't display*/
    printf("Compressed file.");
    fclose(BMPFile);
    return(1);
}

if (Header.biBitCount != 8)
{
    /*If the file is other than 8-bit dont read.*/
    printf("Not an 8-bit bitmap.");
    fclose(BMPFile);
    return(1);
}

if (Header.biWidth > 320 || Header.biHeight > 200)
{
    /*If its larger than 320*200 dont load.*/
    printf("Size too large.");
    fclose(BMPFile);
    return(1);
}

/*Load the palette info*/
fread(&Palette, 1024, 1, BMPFile);

for (c = 0; c < 255; c++)
    SetDAC(c, Palette[c][2] >> 2, Palette[c][1] >> 2, Palette[c][0] >> 2);

offset = (100 + (Header.biHeight >> 1)) * 320 + 160 - (Header.biWidth >> 1);
lines = 0;
paddedWidth = Header.biWidth & 0xFFFC;

if (Header.biWidth != paddedWidth)
    paddedWidth += 4;

/*Loop for reading lines*/
while (lines < Header.biHeight)
{
    fread(Screen + offset, paddedWidth, 1, BMPFile);
    offset -= 320;
    lines++;
}

fclose(BMPFile);
return(0);
}

int main(int argcount, char *argvalue[])
{
    /* Set up a pointer in vga memory */
    Screen = (char far *) 0xA0000000L;
    VGAScreen();
    LoadBMP();
    getch();
    TextScreen();
    return(0);
}

```