Experiment Number: 3

NAME: **Kartik Banshi Katkar** ROLLNO: 36

CLASS: TY IT A BATCH: 1

DATE OF PERFORMANCE: 21/07/2003

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question:  
  
Write C-language code to extract header information of BMP Image. Verify your result using matlab.**

**Answer:**

#include <stdio.h>

#include <stdint.h>

#include <sys/stat.h>

#include <time.h>

#pragma pack(2)

typedef struct {

    uint16\_t type;                 // Magic identifier: 0x4d42

    uint32\_t fileSize;             // Size of the BMP file in bytes

    uint32\_t reserved;             // Reserved

    uint32\_t dataOffset;           // Offset of image data relative to the file's start

    uint32\_t headerSize;           // Size of the BMP header in bytes

    int32\_t width;                 // Width of the image in pixels

    int32\_t height;                // Height of the image in pixels

    uint16\_t planes;               // Number of color planes (must be 1)

    uint16\_t bitsPerPixel;         // Number of bits per pixel

    uint32\_t compression;          // Type of compression being used

    uint32\_t imageSize;            // Size of the raw image data

    int32\_t xPixelsPerMeter;       // Horizontal resolution of the image (pixels per meter)

    int32\_t yPixelsPerMeter;       // Vertical resolution of the image (pixels per meter)

    uint32\_t totalColors;          // Number of colors in the color palette

    uint32\_t importantColors;      // Number of important colors (0 means all colors are important)

} BitmapHeader;

void printFormattedTime(time\_t rawtime) {

    struct tm \*info;

    char buffer[80];

    info = localtime(&rawtime);

    strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M:%S", info);

    printf("FileModDate: %s\n", buffer);

}

void printFileSize(const char \*filename) {

    struct stat st;

    if (stat(filename, &st) == 0)

        printf("FileSize: %lld bytes\n", (long long)st.st\_size);

    else

        printf("Failed to retrieve file size.\n");

}

const char \*getFormatString(uint16\_t type) {

    if (type == 0x4D42)

        return "BMP";

    else

        return "Unknown";

}

const char \*getColorTypeString(uint16\_t bitsPerPixel) {

    if (bitsPerPixel == 24)

        return "True Color";

    else

        return "Unknown";

}

int main() {

    char str[20];

    printf("Enter filename or path: ");

    scanf("%s", &str);

    const char \*filename = str;

    FILE \*file = fopen(filename, "rb");

    if (file == NULL) {

        printf("Failed to open the image file.\n");

        return 1;

    }

    // Read the BMP header

    BitmapHeader header;

    fread(&header, sizeof(BitmapHeader), 1, file);

    // Check if the input image is a BMP file

    if (header.type != 0x4D42) {

        printf("The input file is not a BMP image.\n");

        fclose(file);

        return 1;

    }

    // Print additional file details

    printf("Filename: %s\n", filename);

    printFileSize(filename);

    printFormattedTime(time(NULL));

    printf("Format: %s\n", getFormatString(header.type));

    printf("Width: %d pixels\n", header.width);

    printf("Height: %d pixels\n", header.height);

    printf("BitDepth: %d\n", header.bitsPerPixel);

    printf("ColorType: %s\n", getColorTypeString(header.bitsPerPixel));

    printf("FormatSignature: %u\n", header.type);

    printf("RedMask: %08X\n", 0xFF0000);

    printf("GreenMask: %08X\n", 0x00FF00);

    printf("BlueMask: %08X\n", 0x0000FF);

    printf("ImageDataOffset: %u\n", header.dataOffset);

    printf("BitmapHeaderSize: %u\n", header.headerSize);

    printf("NumPlanes: %d\n", header.planes);

    printf("CompressionType: %u\n", header.compression);

    printf("BitmapSize: %u\n", header.imageSize);

    printf("HorzResolution: %d pixels/meter\n", header.xPixelsPerMeter);

    printf("VertResolution: %d pixels/meter\n", header.yPixelsPerMeter);

    printf("NumColorsUsed: %u\n", header.totalColors);

    fclose(file);

    return 0;

}

**Output:   
  
PS D:\kartik\SY IT SEM - II\IPAC> cd "d:\kartik\SY IT SEM - II\IPAC\" ; if ($?) { gcc bmp.c -o bmp } ; if ($?) { .\bmp }**

**Enter filename or path: kakashibmp.bmp**

**Filename: kakashibmp.bmp**

**FileSize: 152290 bytes**

**FileModDate: 2023-07-27 16:47:26**

**Format: BMP**

**Width: 190 pixels**

**Height: 266 pixels**

**BitDepth: 24**

**ColorType: True Color**

**FormatSignature: 19778**

**RedMask: 00FF0000**

**GreenMask: 0000FF00**

**BlueMask: 000000FF**

**ImageDataOffset: 138**

**BitmapHeaderSize: 124**

**NumPlanes: 1**

**CompressionType: 0**

**BitmapSize: 152152**

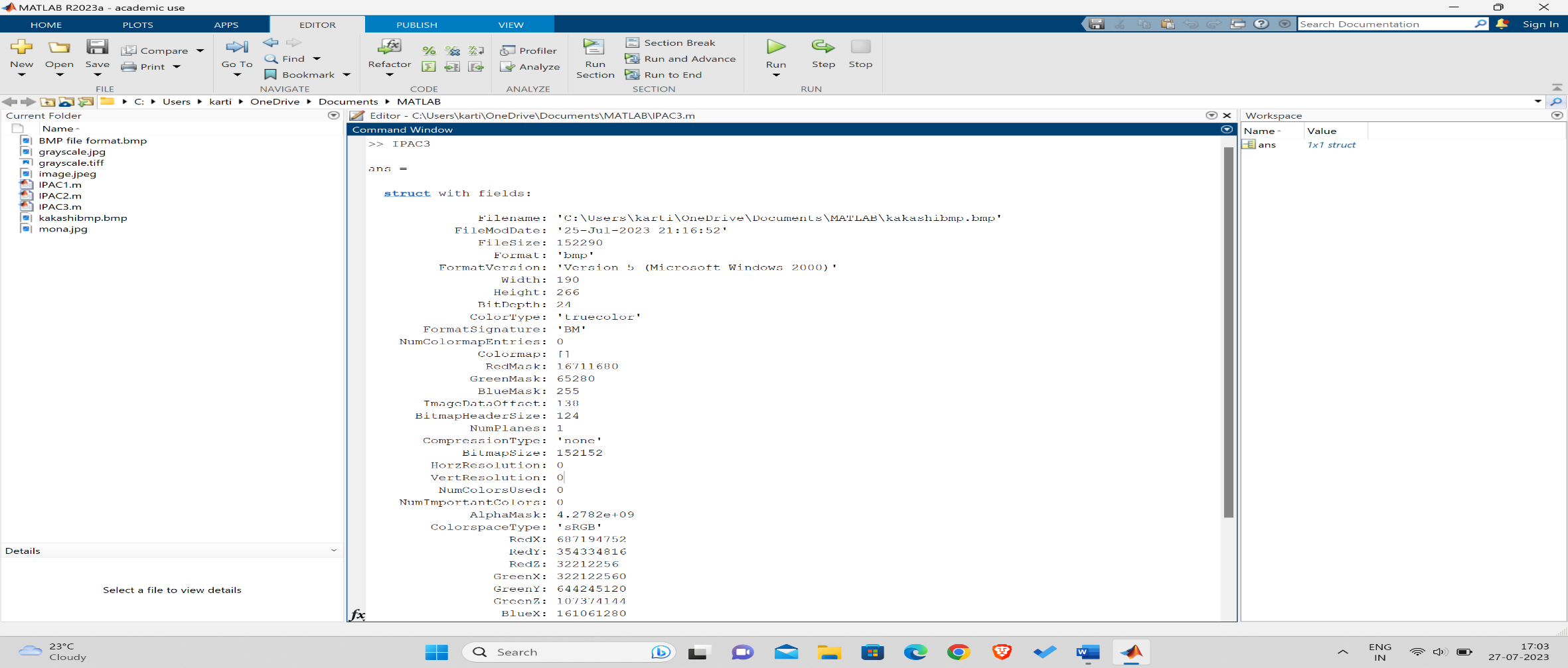
**HorzResolution: 0 pixels/meter**

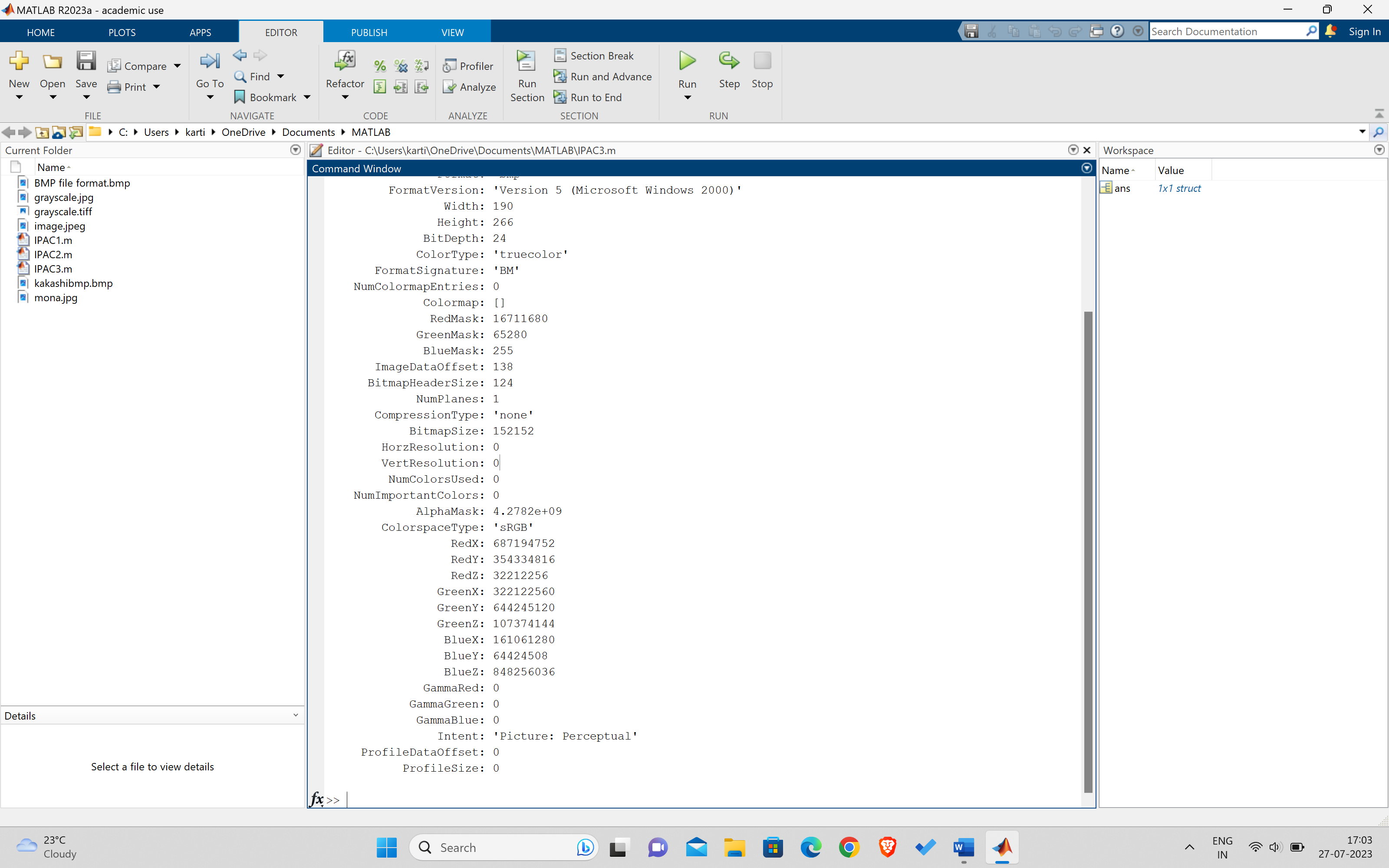
**VertResolution: 0 pixels/meter**

**NumColorsUsed: 0**

**PS D:\kartik\SY IT SEM - II\IPAC>**

**Matlab code:**imfinfo("kakashibmp.bmp")

**Output:  
  
**

****