MULTIMEDIA ASSIGNMENT

**QUESTION 1**

Conditions

Camera Focal Length=f=50mm

Vertical Column= 12 meters

Object Distance=u=95m

U=95m=95000mm’

-Calculate the Image Distance(v)

1/v+1/u=1/f

1/v+1/95000=1/50

1/v=1/50-1/9500

v = (1/50-1/95000)^-1

v=50.03mm

Magnification=v/u

M=50.03/9500

1. Image Height=(50.03/95000)\*12000

=6.32

1. Number of pixels

Inches=25.4

Camera Resolution=300 dots per inch

=(300/25.4)\*6.32

=74.64

(b) Calculate Pixels number

=640\*480

=307200

1. Consider if an 8 bit color is use for each color

Each pixel per bit=8\*3

=24 bits

Total=307200\*24

=7372800 bit

=7372800/8 byte

=921600 bytes

1. Consider if 10 bits is used for color

Bits for each pixel =10\*3=30 bits

Total=307200\*30

=9216000 bit

=9216000/8 byte

=1152000 byte

1. Consider if 12 bits is used for color

Bits per pixel=12\*3=36 bits

Total= 307200\*36 bits

=11059200 bits

=11059200/8 byte

=1382400 byte

1. Consider if 14 bits are used for color

Bits per pixel=14\*3=42 bit

Total=307200\*42 bit

=12902400 bit

=12902400/8 bit

=16120800 byte

**QUESTION 2**

Code for the Program

import ij.\*;

import ij.process.\*;

import ij.plugin. iter.\*;

public class Count\_Color\_Freq implements PluginFilter{

ImagePlus imp;

Public int setup( String arg, ImagePlus imp {

this.imp=imp;

return DOES\_RGB+NO\_UNDO+NO\_CHANGES+DOES\_STACKS;

)

public void run(ImageProcessor ip){

green=0;

red=0;

blue=0;

white=0;

black=0;

int rgb[]=new int[3];

for(y=0;y<ip.getHeight[];y++){

for(x=0;x<ip.getWidth[];x++){

v=ip.getPixel(x,y);

r=(v>>16)&0xff;// extract red byte (bits 23-17)

g=(v>>8)&0xff; // extract green bytes (bits 15-8)

b=v&0xff; // Extract blue bytes (bits 7-0)

//or use this

/\*

ip.getPixel(X,Y,rgb);

r=rgb[0]; g=rgb[1]; b=rgb[2];

\*/

if(r==255 && g=0 && b=0) red++;

else if (r ==0 && g ==255 && b ==0) green++;

else if (r ==0 && g ==0 && b ==255) blue++;

else if (r ==0 && g ==0 && b ==0) black++;

else if (r ==255 && g ==255 && b ==255) white++;

}

}

showMessage("Green = " + green +

", Red = " + red +

", Blue = " + blue +

", Black = " + black +

", White = " + white);

} }

**QUESTION 3**

Background

Histograms are integral part in image editing and in most image editing software the technology is highly used.The following code illustrates the use of histograms in image editing.

Code for The Java-Plugin

public static int max(int[]arr){ int maxValue = arr[0];

for ( int i=1; i < arr.length; i++ ){

if (arr[i] > maxValue){

maxValue = arr[i];

} }

return maxValue;

}

public static int randomInteger(int a, int b){; int randomNum;

randomNum = a+(int)(Math.random() \* ((b-a)+1));

return randomNum;

}

public static void drawHistogram(int[] arr){

for ( int i=0; i<arr.length; i++ ){

System.out.print((i\*10+1)+"-"+(i\*10+10)+":"+"\t");

for (int j=0; j<arr[i]; j++) System.out.print("\*");

System.out.println();

}

}

public static void doSingleTest(int[] arr, int num, int range){

for (int i=1; i<=num; i++){

int random = randomInteger(0,range);

arr[random]++;

} }

public static void doPairsTesting(int[] arr, int num, int range){

}

public static void main(String[] args) {

int test[] = new int[]{1,2,3,4,6,11,7};

System.out.println("mathod1 = "+ max(test));

System.out.println("mathod2 = "+randomInteger(1,20));

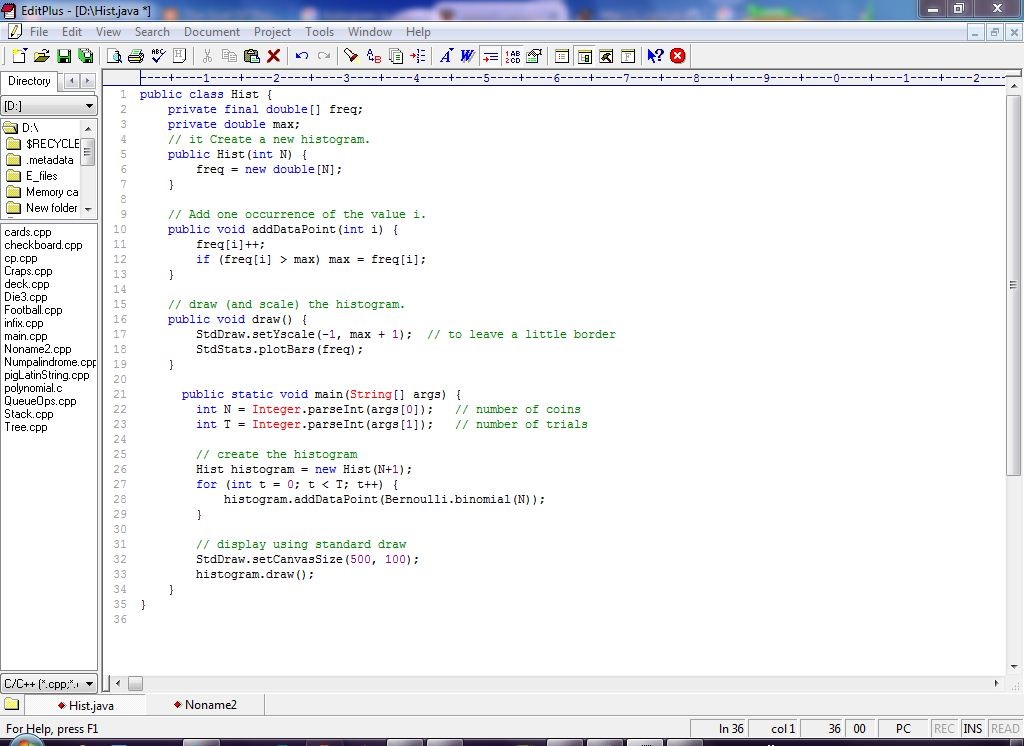
drawHistogram(test);

doSingleTest(test,1,5);

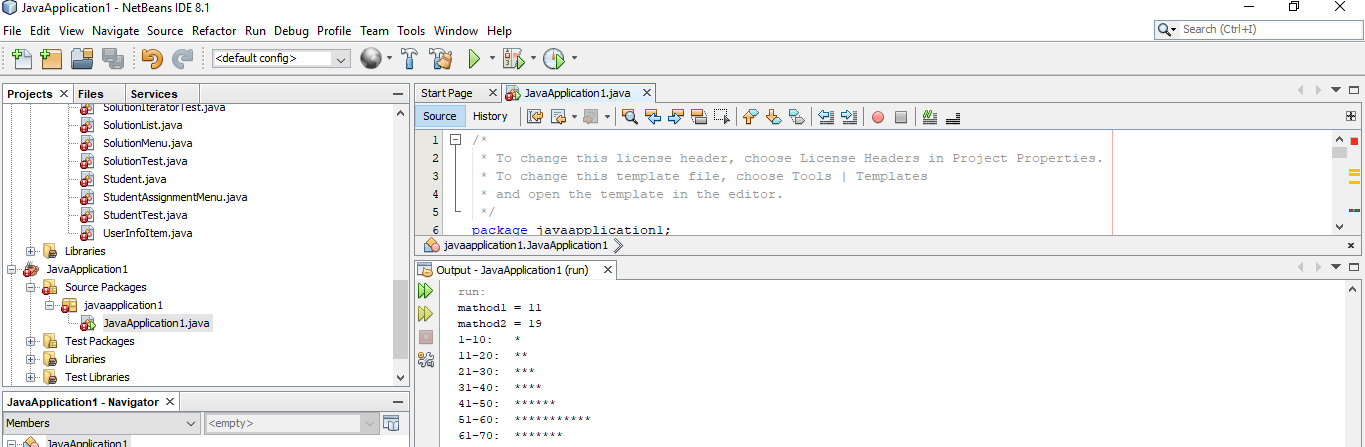
System.out.println("mathod4 = "+Arrays.toString(test));

}

Program Output



Output Run

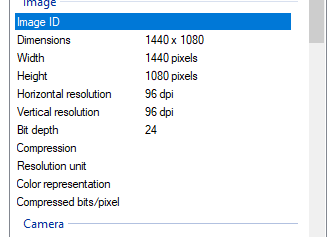


**QUESTION 4**

Image Screenshot



Image Details



I=imread('autumn.tif');

R=imhist(I(:,:,1));

G=imhist(I(:,:,2));

B=imhist(I(:,:,3));

B=imhist(l(:,:,3))

figure

plot(R,'r')

hold on,

plot(G,'g')

plot(B,'b') legend(' Red channel','Green channel','Blue channel');

hold off

(b)

Histograms depicts problems that originate during image acquisition

Exposure, contrast, dynamic range

Histograms can be used to detect a wide range of image defects: saturation, spikes and gaps, impact of image compression

(c)

gammaCorrection = 1 / gamma

colour = GetPixelColour(x, y)

newRed = 255 \* (Red(colour) / 255) ^ gammaCorrection

newGreen = 255 \* (Green(colour) / 255) ^ gammaCorrection

newBlue = 255 \* (Blue(colour) / 255) ^ gammaCorrection

PutPixelColour(x, y) = RGB(newRed, newGreen, newBlue)

(d) For most downloaded images their image visualization has a high fidelity.