

NAME:- SHAHZAD MUSTAFA KAMBOH.

ROLL NO:- 2K18/ELE/102.

NAME:- SADAM HUSSAIN SAMEJO.

ROLL NO:- 2K18/ELE/89.

CLASS: BS-ELECTRONICS P-IV

SUBJECT: COMPUTER VERSION AND IMAGE PROCESSING

SUMBIT: DR.SANDAR ALI

PROGRAM CODE: Using this program a 24-bit bitmap image file can be converted into 8-bit bitmap image file with using process the change of colour information is lost resulting in a grayscale image to exchange information to image.

Ans:

The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Color image processing is divided into two major areas: full-color and pseudo-color processing.

=> The human visual system can distinguish hundreds of thousands of different colour shades and intensities, but only around 100 shades of grey. Therefore, in an image, a great deal of extra information may be contained in the

colour, and this extra information can then be used to simplify image analysis, e.g. object identification and extraction based on colour.

=> A colour image is consist of image is gernal to pixel is given by changing of image effect the color

```
#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<string.h>

#include<stdio.h>

char filename1[25],filename2[25];

int rgbtogley()

{

char c;

int m,n;

ifstream fin(filename1,ios::in|ios::binary); //open input file

if(!fin)

return 0;

ofstream fout (filename2,ios::out::ios::binary); // create output file

for(int i=0;i<54;i++) //copy header from input file to output file {

fin.read((char*)&c,sizeof(c));

fout.write((char*)&c,sizeof(c));

}

for (int i1=0;i1<256;i1++) //create palette in output file
```

```
{  
    c=i1;  
  
    fout.write((char*)&c,sizeof(c));  
    fout.write((char*)&c,sizeof(c));  
    fout.write((char*)&c,sizeof(c));  
  
    c=0;  
  
    fout.write((char*)&c,sizeof(c));  
}  
  
while (!fin.eof()) //read the three values and find average  
{  
    m=0;  
  
    fin.read((char*)&c,sizeof(c));  
  
    n=c;  
  
    if(n<0)  
        n+=256;  
  
    m+=n;  
  
    fin.read((char*)&c,sizeof(c));  
  
    n=c;  
  
    if(n<0)  
        n+=256;  
  
    m+=n;  
  
    fin.read((char*)&c,sizeof(c));  
  
    n=c;  
  
    if(n<0)  
        n+=256;
```

```

m+=n;

m=n/3;

if(m>255)

m=255;

else if(m<0)

m=0;

c=m;

fout.write((char*)&c,sizeof(c)); //write image data to output file

}

fout.seekp(20); // number of bits per pixel

c=0;

fout.write((char*)&c,sizeof(c));

fout.seekp(11); // offset to start image data

c=4;

fout.write((char*)&c,sizeof(c));

fout.close();

fin.close ();

return 1;

}

void main()

{

clrscr();

int r;

cout<<"\nconvert 24 bit bmpfile to 8bit bmp file\n\n";

cout<<"/nNo need to specify .bmpfile extention along with file name\n";

```

```
cout<<"\nEnter required output file name: ";  
gets (file name1);  
cout<<"\nEnter source output file name: ";  
gets (file name2);  
strcat (filename1, ".bmp");  
strcat (filename2, ".bmp");  
r= regbtogrey();  
if(r==1)  
cout<<"\nconversion successfull..!";  
else  
cout<<"\n\nEnter= file not found..!";  
getch();  
}
```



Fig. 2: Input file (24-bit bitmap)

