Place images in the current folder & run this code. If result.bmp file has reverse image, go ahead.

#include <iostream>

#include <fstream>

using namespace std;

#define HSZIE 1078

#define COLS 200

#define ROWS 200

void readFile(string fName, unsigned char h[], unsigned char m[][COLS]){

ifstream in(fName.c\_str(),ios::binary);

in.read((char\*)h,HSZIE);

in.read((char\*)m,ROWS\*COLS);

in.close();

}

void writeFile(string fName, unsigned char h[], unsigned char m[][COLS]){

ofstream out(fName.c\_str(),ios::binary);

out.write((char\*)h,HSZIE);

out.write((char\*)m,ROWS\*COLS);

out.close();

}

void reverse(unsigned char m1[][COLS], unsigned char m2[][COLS]){

int i,j, k, m, n;

for (i=0,m=ROWS-1;i<ROWS;i++,m--)

for (j=0,n=COLS-1;j<COLS;j++, n--)

m2[m][n]=m1[i][j];

}

int main(){

unsigned char h[HSZIE], m1[ROWS][COLS], m2[ROWS][COLS];

readFile("mm01.bmp",h, m1);

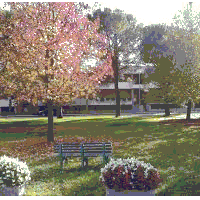
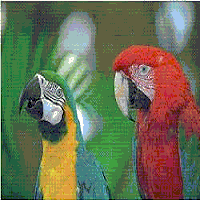
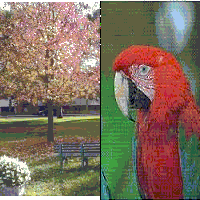
reverse(m1, m2);

writeFile("result.bmp",h, m3);

return 0;

}

**Task 1.** Write function merge to merge 2 images such that first half is taken from image1 & second half is taken from image2. In main read 2 files and pass to merge function. The resultant image should be like:

|  |  |  |
| --- | --- | --- |
| MM01.BMP | MM02.BMP | result.BMP |

#include <iostream>

#include <fstream>

using namespace std;

#define HSZIE 54

#define COLS 200

#define ROWS 200

//Don't concentrate on read & write functions, just concentrate on reverse function

void readFile(string fName, unsigned char h[], unsigned char m[][COLS][3]){

ifstream in(fName.c\_str(),ios::binary);

in.read((char\*)h,HSZIE);

in.read((char\*)m,ROWS\*COLS\*3);

in.close();

}

void writeFile(string fName, unsigned char h[], unsigned char m[][COLS][3]){

ofstream out(fName.c\_str(),ios::binary);

out.write((char\*)h,HSZIE);

out.write((char\*)m,ROWS\*COLS\*3);

out.close();

}

void removeRed(unsigned char m1[][COLS][3], unsigned char m2[][COLS][3]){

int i,j, k, m, n;

for (i=0;i<ROWS;i++)

for (j=0;j<COLS;j++){

m2[i][j][0]=m1[i][j][0];//Assign Green to Green

m2[i][j][1]=m1[i][j][1];//Assign Blue to Blue

m2[i][j][2]=0;//Assign 0 to Red

}

}

int main(){

unsigned char h[HSZIE], m1[ROWS][COLS][3], m2[ROWS][COLS][3];

readFile("im02.bmp",h, m1);

removeRed(m1, m2);

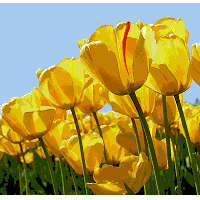
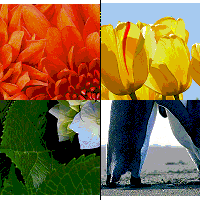
writeFile("result.bmp",h, m2);

return 0;

}

Apply code written for 24-bit color image & see result.

Task 2. Write function to merge 4 images into 1 image. See pictures below to get understanding.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IM01.BMP | IM02.BMP | IM03.BMP | IM04.BMP | result.BMP |