POOJA SINGH C (18BCT035)

COMPUTER TECHNOLOGY

**Problem Statement :**

Implement a median filter from scratch using C++ .  Input should be an Image and the output should be a Linear filtered Image, Neat Documentation is expected with Code, Explanation, Input, and Output Image.

**PROGRAM:**

#include <iostream>

#include <fstream>

#include <sstream>

using namespace std;

void insertionSort(int arr[], int n)

{

int i, key, j;

for (i = 1; i < n; i++)

{

key = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > key)

{

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

int array[2000][2000];

int arr[2000][2000];

int main()

{

int window[9],row = 0, col = 0, numrows = 0, numcols = 0,MAX=0;

ifstream infile("Saltpepper.pgm");

stringstream ss;

string inputLine = "";

getline(infile,inputLine);

if(inputLine.compare("P2") != 0) cerr << "Version error" << endl;

else cout << "Version : " << inputLine << endl;

ss << infile.rdbuf();

ss >> numcols >> numrows >> MAX;

cout << numcols << " columns and " << numrows << " rows" <<endl<<"

Maximium Intesity "<< MAX <<endl;

for(row = 0; row <= numrows; ++row)

array[row][0]=0;

for( col = 0; col<=numcols; ++col )

array[0][col]=0;

for(row = 1; row <= numrows; ++row)

{

for (col = 1; col <= numcols; ++col)

{

ss >> array[row][col];

}

}

for(row = 1; row <= numrows; ++row)

{

for(col = 1; col <= numcols; ++col)

{

//neighbor pixel values are stored in window including this pixel

window[0] = array[row-1][col-1];

window[1] = array[row-1][col];

window[2] = array[row-1][col+1];

window[3] = array[row][col-1];

window[4] = array[row][col];

window[5] = array[row][col+1];

window[6] = array[row+1][col-1];

window[7] = array[row+1][col];

window[8] = array[row+1][col+1];

insertionSort(window,9);

arr[row][col]=window[4];

}

}

ofstream outfile;

outfile.open("Medianfilter.pnm");

outfile<<"P2"<<endl;

outfile<<numcols<<" "<<numrows<<endl;

outfile<<"255"<<endl;

for(row = 1; row <= numrows; ++row)

{

for (col = 1; col <= numcols; ++col)

{

//store resultant pixel values to the output file

outfile << arr[row][col]<<" ";

}

}

outfile.close();

infile.close();

return 0 ;

}