**C++ IMPLEMENTATION OF MEDIAN FILTER ALLGORITHM**:

#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 arr[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)

        {

            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)

        {

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

        }

    }

    outfile.close();

    infile.close();

    return 0 ;

}

INPUT IMAGE:



OUTPUT IMAGE:



EXPLANATION:

* The median filter is also used to preserve edge properties while reducing the noise. Also, the smoothing techniques, like Gaussian blur is also used to reduce noise but it can’t preserve the edge properties.
* The median filter is widely used in digital image processing just because it preserves edge properties. Store the pixel values of input image in an array.
* For each pixel value store all the neighbour pixel value including that cell in a new array (called window).
* Sort the window array.
* Median of window array is used to store output image pixel intensity.
* If there is an odd amount of numbers, the median value is the number that is in the middle, with the same amount of numbers below and above.
* If there is an even amount of numbers in the list, the median is the average of the two middle values.