#include <iostream>

using namespace std;

int partition(int arr[], int start, int end)

{

int pivot = arr[start];

int count = 0;

for (int i = start + 1; i <= end; i++)

{

if (arr[i] <= pivot)

count++;

}

int pivotIndex = start + count;

swap(arr[pivotIndex], arr[start]);

int i = start, j = end;

while (i < pivotIndex && j > pivotIndex)

{

while (arr[i] <= pivot)

{

i++;

}

while (arr[j] > pivot)

{

j--;

}

if (i < pivotIndex && j > pivotIndex)

{

swap(arr[i++], arr[j--]);

}

}

return pivotIndex;

}

void quickSort(int arr[], int start, int end)

{

if (start >= end)

return;

int pivotIndex = partition(arr, start, end);

quickSort(arr, start, pivotIndex - 1);

quickSort(arr, pivotIndex + 1, end);

}

int main(){

int arr[][3] = {{1,2,3},{1,2,2},{1,1,1},{2,2,2},{2,1,1},{2,1,2},{3,3,3},{3,2,2},{3,1,1}};

cout << "Before sorting" << endl;

int len= sizeof(arr)/sizeof(arr[0]);

for (int i=0 ; i<len ; i++){

for (int j=0 ; j<3 ; j++){

cout << arr[i][j] << " ";

}

cout << endl;

}

cout << "After sorting" << endl;

int arr2[len];

for (int i=0 ; i<len ; i++){

arr2[i]=arr[i][0]\*100+arr[i][1]\*10+arr[i][2];

}

quickSort(arr2, 0, len-1);

for (int i=0 ; i<len ; i++){

arr[i][0]=arr2[i]/100;

arr[i][1]=(arr2[i]%100)/10;

arr[i][2]=arr2[i]%10;

}

for (int i=0 ; i<len ; i++){

for (int j=0 ; j<3 ; j++){

cout << arr[i][j] << " ";

}

cout << endl;

}

}

A picture containing table

Description automatically generated