

```
function Runden(n,Zahl) {
    var Zehnerpotenz = Math.pow(10,n);

    return(Math.round(Zahl * Zehnerpotenz) / Zehnerpotenz);
}

function Summe_Array(Array,StellenanzahlnachSeparator) {
    var Summe = 0;

    for(var i = 0;i < Array.length;i++) {
        Summe += Array[i];
    }

    return Runden(StellenanzahlnachSeparator,Summe);
}

function Summe_Array_Array(Matrix,StellenanzahlnachSeparator) {
    var Summe = new Array(Matrix.length);

    for(var i = 0;i < Summe.length;i++) {
        Summe[i] = Summe_Array(Matrix[i],StellenanzahlnachSeparator);
    }

    return Summe;
}

function vertauschen(Array,i,j) {
    var hilf = Array[i];

    Array[i] = Array[j];

    Array[j] = hilf;

    return Array;
}

var HauptarrayNebenarray = new Array(0,0);

function Quicksort(linkerIndex,rechterIndex,Sortierungsart) {
    var Hauptarray = HauptarrayNebenarray[0];

    var Nebenarray = HauptarrayNebenarray[1];

    var Trenner = Hauptarray[Math.floor((linkerIndex + rechterIndex) / 2)];

    var linkerLaufindex = linkerIndex;

    var rechterLaufindex = rechterIndex;

    while(true) {
        switch(Sortierungsart) {
            case 1: {
                while(Hauptarray[linkerLaufindex] < Trenner) {
                    linkerLaufindex++;
                }

                while(Trenner < Hauptarray[rechterLaufindex]) {
```

```
        rechterLaufindex--;
    }

    break;
}

case 2: {
    while(Hauptarray[linkerLaufindex] > Trenner) {
        linkerLaufindex++;
    }

    while(Trenner > Hauptarray[rechterLaufindex]) {
        rechterLaufindex--;
    }

    break;
}
}

if(linkerLaufindex > rechterLaufindex) {
    break;
}

Hauptarray = vertauschen(Hauptarray,linkerLaufindex,rechterLaufindex);

Nebenarray = vertauschen(Nebenarray,linkerLaufindex,rechterLaufindex);

linkerLaufindex++;

rechterLaufindex--;

if(linkerLaufindex > rechterLaufindex) {
    break;
}
}

if(linkerIndex < rechterLaufindex) {
    HauptarrayNebenarray = new Array(Hauptarray,Nebenarray);

    Quicksort(linkerIndex,rechterLaufindex,Sortierungsart);
}

if(linkerLaufindex < rechterIndex) {
    HauptarrayNebenarray = new Array(Hauptarray,Nebenarray);

    Quicksort(linkerLaufindex,rechterIndex,Sortierungsart);
}
}

function sortieren_von_gross_nach_klein_Matrix(Hauptarray,Nebenarray) {
    HauptarrayNebenarray = new Array(Hauptarray,Nebenarray);

    Quicksort(0,Hauptarray.length - 1,2);

    return HauptarrayNebenarray;
}
```