void bubblesort( arr, length){

int I,j;

for(0 to end){

i++;

for(do loop from 0 to end – i){

group two data

if(front data is bigger than behind data) {

swap(front,behind)

//then,largest data will be locate at the back

}

}

}

}

//C CODE

void swap(int\* num1, int\* num2) { //swap two datas

int temp;

temp = \*num1;

\*num1 = \*num2;

\*num2 = temp;

}

// program your bubble sorting algorithm

void bubbleSorting(int\* original\_list, int numofkeys)

{

for (int i = 0; i < numofkeys - 1; i++) {

for (int j = 0; j < numofkeys - i - 1; j++) {

if (original\_list[j] > original\_list[j + 1]) {

swap(&original\_list[j], &original\_list[j+1]);

};

};

};

}

////////////////////////////////////////////////////////////////

void quickSort(int data, int start, int end)

int key = start; //set pivot to start index

while (start pivot meets end pivot) {

if start pivot`s data <= key`s data , stop moving toward

if go toward , start++;

//if end pivot`s data >= key`s data , stop moving behind

If go behind,end++;

if (start pivot > end pivot ) {

swap(&data[end], &data[key]);

}

else {

swap(&data[start], &data[end]);

}

}

Quicksort(data,0,end); //separate array based on the meeting point(start,end)

Quicksort(data,end+ 1,data.length-1);

//C CODE

void swap(int\* num1, int\* num2) { //swap two datas

int temp;

temp = \*num1;

\*num1 = \*num2;

\*num2 = temp;

}

// program your second sorting algorithm

void yourSecondSorting(int\* data, int start, int end) //I will use quick sort

{

if (start >= end) //if array`s length is 1,just done

return;

int key = start; //key is first element

int i = start + 1, j = end, temp;

while (i <= j) { //while start pivot meets end pivot

while (i <= end && data[i] <= data[key]) { //if start pivot`s data <= key`s data , stop moving toward

i++;

}

while (j > start && data[j] >= data[key]) { //if end pivot`s data >= key`s data , stop moving behind

j--;

}

if (i > j) {

swap(&data[j], &data[key]);

}

else {

swap(&data[i], &data[j]);

}

}

yourSecondSorting(data, start, j - 1); //separate array based on the meeting point(start pivot,end pivot) and do sort recursively

yourSecondSorting(data, j + 1, end);

}