Void merge(int data[], int start, int end);

Void mergeSort(int data[], int start, int end){

If (start >= end){

Return;

}

Int mid = (start + end)/2;

MergeSort(data, start, mid);

mergeSort(data, mid + 1, end);

merge(data, start, mid, end);

}

Void merge(int data[], int start, int end){

Int left[mid – start + 1];

Int right[end – mid];

Int \*lp = left;

Int \*rp = right;

Int \*dp = &data[0];

For (int I = 0; i < mid – start + 1; i++){

Left[i]= data[i];

}

For (int I = 0; I < end – mid; i++){

Right[i] = data[mid + 1+ i];

}

For (int I = 0; I < end – start + 1; i++){

If (lp < &lp[mid – start]) {

\*dp = \*lp;

}

Else if (rp > &right[end – mid – 1]){

\*dp = \*lp;

Lp++;

}

Else if (\*lp < \*rp){

\*dp = \*lp;

}

Else {

\*dp = \*rp;

Rp++;

}

dp==;

}

}