

#include <stdio.h>

#define N 10

void input(int a[], int n);

void output(int a[], int n);

void sort(int a[], int n);

int main(void)

{

int a[N];

input(a, N);

sort(a, N);

output(a, N);

return 0;

}

void input(int a[], int n)

{

int i;

printf("Input:\n");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

}

void output(int a[], int n)

{

int i;

printf("Output:\n");

for (i = 0; i < n; i++)

{

printf("%d ", a[i]);

}

printf("\n");

}

void sort(int a[], int n)

{

int i, j, k, temp;

for (i = 0; i < 3; i++) /\*注：该排序算法跑三趟，将3改为n即为完全排序算法\*/

{

k = i;

/\*\*\*\*\*\*start\*\*\*\*\*\*/

for (j = i + 1; j < n; j++)

{

if (a[k] > a[j])

{

k = j;

}

}

if (i != k)

{

temp = a[i];

a[i] = a[k];

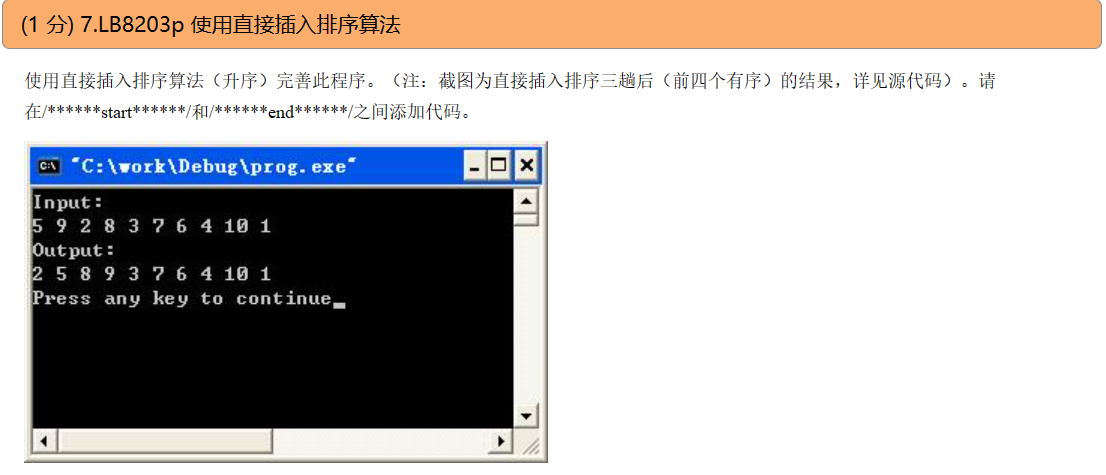
a[k] = temp;

}

/\*\*\*\*\*\*end\*\*\*\*\*\*/

}

}



#include <stdio.h>

#define N 10

void input(int a[], int n);

void output(int a[], int n);

void sort(int a[], int n);

int main(void)

{

int a[N];

input(a, N);

sort(a, N);

output(a, N);

return 0;

}

void input(int a[], int n)

{

int i;

printf("Input:\n");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

}

void output(int a[], int n)

{

int i;

printf("Output:\n");

for (i = 0; i < n; i++)

{

printf("%d ", a[i]);

}

printf("\n");

}

void sort(int a[], int n)

{

int i, j, temp;

for (i = 1; i < 4; i++) /\*注：该排序算法跑三趟，将4改为n即为完全排序算法\*/

{

/\*\*\*\*\*\*start\*\*\*\*\*\*/

temp = a[i];

for (j = i - 1; j >= 0; j--)

{

if (a[j] > temp)

{

a[j+1] = a[j];

}

else

{

break;

}

}

a[j+1] = temp;

/\*\*\*\*\*\*end\*\*\*\*\*\*/

a[j+1] = temp;

}

}



#include <stdio.h>

#define N 10

void input(int a[], int n);

void output(int a[], int n);

int sort(int a[], int low, int high);

void quicksort(int a[], int low, int high);

int main(void)

{

int a[N];

input(a, N);

quicksort(a, 0, N);

output(a, N);

return 0;

}

void input(int a[], int n)

{

int i;

printf("Input:\n");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

}

void output(int a[], int n)

{

int i;

printf("Output:\n");

for (i = 0; i < n; i++)

{

printf("%d ", a[i]);

}

printf("\n");

}

void quicksort(int a[], int low, int high)

{

int pivotloc; /\*枢轴下标\*/

/\*注：该排序算法跑一趟。如果将下面3行注释取消，即为完全排序算法\*/

//if (low < high)

{

pivotloc = sort(a, low, high);

//quicksort(a, low, pivotloc - 1); /\*递归将前半部分元素继续快速排序\*/

//quicksort(a, pivotloc + 1, high);

}

}

int sort(int a[], int low, int high)

{

int pivotkey;

pivotkey = a[low];

/\*\*\*\*\*\*start\*\*\*\*\*\*/

无

/\*\*\*\*\*\*end\*\*\*\*\*\*/

}



#include <stdio.h>

#define N 10

void input(int a[], int n);

void output(int a[], int n);

void sort(int a[], int n);

int incre[3] = {5, 3, 1}; /\* 增量序列 \*/

int main(void)

{

int a[N];

input(a, N);

sort(a, N);

output(a, N);

return 0;

}

void input(int a[], int n)

{

int i;

printf("Input:\n");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

}

void output(int a[], int n)

{

int i;

printf("Output:\n");

for (i = 0; i < n; i++)

{

printf("%d ", a[i]);

}

printf("\n");

}

void sort(int a[], int n)

{

int i, j, k, d, temp;

for (k = 0; k < 2; k++) /\*注：该排序算法跑两趟，将2改为3即为完全排序算法\*/

{

d = incre[k]; //d为增量值

/\*\*\*\*\*\*start\*\*\*\*\*\*/

for (i = d; i < n; i++)

{

temp = a[i];

for (j = i - d; j >= 0 && a[j] > temp; j -= d)

{

a[j+d] = a[j];

}

a[j+d] = temp;

}

/\*\*\*\*\*\*end\*\*\*\*\*\*/

}

}



#include <stdio.h>

#define N 10

void input(int a[], int n);

void output(int a[], int n);

void sort(int a[], int n);

int main(void)

{

int a[N];

input(a, N);

sort(a, N);

output(a, N);

return 0;

}

void input(int a[], int n)

{

int i;

printf("Input:\n");

for (i = 0; i < n; i++)

{

scanf("%d", &a[i]);

}

}

void output(int a[], int n)

{

int i;

printf("Output:\n");

for (i = 0; i < n; i++)

{

printf("%d ", a[i]);

}

printf("\n");

}

void sort(int a[], int n)

{

int i, j, temp;

for (i = 0; i < 3; i++) /\*注：该排序算法跑三趟，将3改为n即为完全排序算法\*/

{

/\*\*\*\*\*\*start\*\*\*\*\*\*/

for (j = 0; j < n - i - 1; j++)

{

if (a[j + 1] < a[j])

{

temp = a[j+1];

a[j+1] = a[j];

a[j] = temp;

}

}

/\*\*\*\*\*\*end\*\*\*\*\*\*/

}

}