**1**

#include <stdio.h>

//void simple\_sort(int [],int);

void swap\_data(int [],int,int);

void print\_data(int,int [],int);

void bubble\_sort(int [],int);

int main(){

int a[] ={9,5,2,7,1,0,6,4,3,8};

int n =10;

print\_data(0,a,n);

printf("\n");

bubble\_sort(a,n);

return 0;

}

void swap\_data(int a[],int i,int j){

int temp;

temp = a[i];

a[i]=a[j];

a[j]=temp;

}

/\*void simple\_sort(int a[],int n){

int i,j;

int times = 1;

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

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

if(a[j]<a[i]){

swap\_data(a,i,j);

}

}

print\_data(times,a,n);

times++;

}

}\*/

void print\_data(int times,int a[],int n){

int i;

printf("%d th: [ ",times);

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

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

}

printf("]\n");

}

void bubble\_sort(int a[],int n){

int i,j;

int times=1;

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

for(j=n-1;j>i;j--){

if(a[j-i]>a[j]){

swap\_data(a,j-1,j);

}

print\_data(times,a,n);

times++;

}

printf("\n");

}

}

実行結果

0 th: [ 9 5 2 7 1 0 6 4 3 8 ]

1 th: [ 9 5 2 7 1 0 6 4 3 8 ]

2 th: [ 9 5 2 7 1 0 6 4 3 8 ]

3 th: [ 9 5 2 7 1 0 6 4 3 8 ]

4 th: [ 9 5 2 7 1 0 6 4 3 8 ]

5 th: [ 9 5 2 7 1 0 6 4 3 8 ]

6 th: [ 9 5 2 7 1 0 6 4 3 8 ]

7 th: [ 9 5 2 7 1 0 6 4 3 8 ]

8 th: [ 9 5 2 7 1 0 6 4 3 8 ]

9 th: [ 9 5 2 7 1 0 6 4 3 8 ]

10 th: [ 9 5 2 7 1 0 6 4 3 8 ]

11 th: [ 9 5 2 7 1 0 6 3 4 8 ]

12 th: [ 9 5 2 7 1 0 3 6 4 8 ]

13 th: [ 9 5 2 7 1 0 3 6 4 8 ]

14 th: [ 9 5 2 7 0 1 3 6 4 8 ]

15 th: [ 9 5 2 0 7 1 3 6 4 8 ]

16 th: [ 9 5 0 2 7 1 3 6 4 8 ]

17 th: [ 9 0 5 2 7 1 3 6 4 8 ]

18 th: [ 9 0 5 2 7 1 3 6 4 8 ]

19 th: [ 9 0 5 2 7 1 3 6 4 8 ]

20 th: [ 9 0 5 2 7 1 3 6 4 8 ]

21 th: [ 9 0 5 2 7 3 1 6 4 8 ]

22 th: [ 9 0 5 2 7 3 1 6 4 8 ]

23 th: [ 9 0 5 2 7 3 1 6 4 8 ]

24 th: [ 9 0 5 2 7 3 1 6 4 8 ]

25 th: [ 9 0 5 2 7 3 1 6 4 8 ]

26 th: [ 9 0 5 2 7 3 1 6 4 8 ]

27 th: [ 9 0 5 2 7 3 6 1 4 8 ]

28 th: [ 9 0 5 2 7 3 6 1 4 8 ]

29 th: [ 9 0 5 2 3 7 6 1 4 8 ]

30 th: [ 9 0 5 2 3 7 6 1 4 8 ]

31 th: [ 9 0 5 2 3 7 6 1 4 8 ]

32 th: [ 9 0 5 2 3 7 6 1 4 8 ]

33 th: [ 9 0 5 2 3 7 1 6 4 8 ]

34 th: [ 9 0 5 2 3 1 7 6 4 8 ]

35 th: [ 9 0 5 2 3 1 7 6 4 8 ]

36 th: [ 9 0 5 2 3 1 7 6 4 8 ]

37 th: [ 9 0 5 2 3 1 7 6 4 8 ]

38 th: [ 9 0 5 2 3 1 7 6 4 8 ]

39 th: [ 9 0 5 2 3 1 7 6 4 8 ]

40 th: [ 9 0 5 2 3 1 7 6 4 8 ]

41 th: [ 9 0 5 2 3 1 7 4 6 8 ]

42 th: [ 9 0 5 2 3 1 7 4 6 8 ]

43 th: [ 9 0 5 2 3 1 7 4 6 8 ]

44 th: [ 9 0 5 2 3 1 7 4 6 8 ]

45 th: [ 9 0 5 2 3 1 7 4 6 8 ]

#include <stdio.h>

//void simple\_sort(int [],int);

void swap\_data(int [],int,int);

void print\_data(int,int [],int);

void bubble\_sort(int [],int);

int main(){

int a[] ={9,5,2,7,1,0,6,4,3,8};

int n =10;

print\_data(0,a,n);

printf("\n");

bubble\_sort(a,n);

return 0;

}

void swap\_data(int a[],int i,int j){

int temp;

temp = a[i];

a[i]=a[j];

a[j]=temp;

}

/\*void simple\_sort(int a[],int n){

int i,j;

int times = 1;

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

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

if(a[j]<a[i]){

swap\_data(a,i,j);

}

}

print\_data(times,a,n);

times++;

}

}\*/

void print\_data(int times,int a[],int n){

int i;

printf("%d th: [ ",times);

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

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

}

printf("]\n");

}

void bubble\_sort(int a[],int n){

int i,j;

int times=1;

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

for(j=n-1;j>i;j--){

if(a[j-i]<a[j]){

swap\_data(a,j-1,j);

}

print\_data(times,a,n);

times++;

}

printf("\n");

}

}

実行結果

0 th: [ 9 5 2 7 1 0 6 4 3 8 ]

1 th: [ 9 5 2 7 1 0 6 4 3 8 ]

2 th: [ 9 5 2 7 1 0 6 4 3 8 ]

3 th: [ 9 5 2 7 1 0 6 4 3 8 ]

4 th: [ 9 5 2 7 1 0 6 4 3 8 ]

5 th: [ 9 5 2 7 1 0 6 4 3 8 ]

6 th: [ 9 5 2 7 1 0 6 4 3 8 ]

7 th: [ 9 5 2 7 1 0 6 4 3 8 ]

8 th: [ 9 5 2 7 1 0 6 4 3 8 ]

9 th: [ 9 5 2 7 1 0 6 4 3 8 ]

10 th: [ 9 5 2 7 1 0 6 4 8 3 ]

11 th: [ 9 5 2 7 1 0 6 8 4 3 ]

12 th: [ 9 5 2 7 1 0 8 6 4 3 ]

13 th: [ 9 5 2 7 1 8 0 6 4 3 ]

14 th: [ 9 5 2 7 8 1 0 6 4 3 ]

15 th: [ 9 5 2 8 7 1 0 6 4 3 ]

16 th: [ 9 5 8 2 7 1 0 6 4 3 ]

17 th: [ 9 8 5 2 7 1 0 6 4 3 ]

18 th: [ 9 8 5 2 7 1 0 6 4 3 ]

19 th: [ 9 8 5 2 7 1 0 4 6 3 ]

20 th: [ 9 8 5 2 7 1 4 0 6 3 ]

21 th: [ 9 8 5 2 7 1 4 0 6 3 ]

22 th: [ 9 8 5 2 7 1 4 0 6 3 ]

23 th: [ 9 8 5 7 2 1 4 0 6 3 ]

24 th: [ 9 8 5 7 2 1 4 0 6 3 ]

25 th: [ 9 8 5 7 2 1 4 0 6 3 ]

26 th: [ 9 8 5 7 2 1 4 6 0 3 ]

27 th: [ 9 8 5 7 2 1 6 4 0 3 ]

28 th: [ 9 8 5 7 2 1 6 4 0 3 ]

29 th: [ 9 8 5 7 2 1 6 4 0 3 ]

30 th: [ 9 8 5 7 2 1 6 4 0 3 ]

31 th: [ 9 8 5 7 2 1 6 4 3 0 ]

32 th: [ 9 8 5 7 2 1 6 3 4 0 ]

33 th: [ 9 8 5 7 2 1 6 3 4 0 ]

34 th: [ 9 8 5 7 2 6 1 3 4 0 ]

35 th: [ 9 8 5 7 2 6 1 3 4 0 ]

36 th: [ 9 8 5 7 2 6 1 3 4 0 ]

37 th: [ 9 8 5 7 2 6 1 3 4 0 ]

38 th: [ 9 8 5 7 2 6 1 3 4 0 ]

39 th: [ 9 8 5 7 2 6 1 3 4 0 ]

40 th: [ 9 8 5 7 2 6 1 3 4 0 ]

41 th: [ 9 8 5 7 2 6 1 3 4 0 ]

42 th: [ 9 8 5 7 2 6 1 3 4 0 ]

43 th: [ 9 8 5 7 2 6 1 3 4 0 ]

44 th: [ 9 8 5 7 2 6 1 3 4 0 ]

45 th: [ 9 8 5 7 2 6 1 3 4 0 ]

**2**

#include <stdio.h>

//void simple\_sort(int [],int);

void swap\_data(int [],int,int);

void print\_data(int,int [],int);

//void bubble\_sort(int [],int);

void insertion\_sort(int [],int);

int main(){

int a[] ={9,5,2,7,1,0,6,4,3,8};

int n =10;

print\_data(0,a,n);

printf("\n");

insertion\_sort(a,n);

return 0;

}

void swap\_data(int a[],int i,int j){

int temp;

temp = a[i];

a[i]=a[j];

a[j]=temp;

}

/\*void simple\_sort(int a[],int n){

int i,j;

int times = 1;

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

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

if(a[j]<a[i]){

swap\_data(a,i,j);

}

}

print\_data(times,a,n);

times++;

}

}\*/

void print\_data(int times,int a[],int n){

int i;

printf("%d th: [ ",times);

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

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

}

printf("]\n");

}

/\*void bubble\_sort(int a[],int n){

int i,j;

int times=1;

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

for(j=n-1;j>i;j--){

if(a[j-i]<a[j]){

swap\_data(a,j-1,j);

}

print\_data(times,a,n);

times++;

}

printf("\n");

}

}\*/

void insertion\_sort(int a[],int n){

int i,j;

int w;

int times =1;

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

w=a[i];

j=i-1;

while(j>=0 && w<a[j]){

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

j--;

print\_data(times,a,n);

times++;

}

a[j+1]=w;

print\_data(times,a,n);

times++;

printf("\n");

}

}

実行結果

0 th: [ 9 5 2 7 1 0 6 4 3 8 ]

1 th: [ 9 9 2 7 1 0 6 4 3 8 ]

2 th: [ 5 9 2 7 1 0 6 4 3 8 ]

3 th: [ 5 9 9 7 1 0 6 4 3 8 ]

4 th: [ 5 5 9 7 1 0 6 4 3 8 ]

5 th: [ 2 5 9 7 1 0 6 4 3 8 ]

6 th: [ 2 5 9 9 1 0 6 4 3 8 ]

7 th: [ 2 5 7 9 1 0 6 4 3 8 ]

8 th: [ 2 5 7 9 9 0 6 4 3 8 ]

9 th: [ 2 5 7 7 9 0 6 4 3 8 ]

10 th: [ 2 5 5 7 9 0 6 4 3 8 ]

11 th: [ 2 2 5 7 9 0 6 4 3 8 ]

12 th: [ 1 2 5 7 9 0 6 4 3 8 ]

13 th: [ 1 2 5 7 9 9 6 4 3 8 ]

14 th: [ 1 2 5 7 7 9 6 4 3 8 ]

15 th: [ 1 2 5 5 7 9 6 4 3 8 ]

16 th: [ 1 2 2 5 7 9 6 4 3 8 ]

17 th: [ 1 1 2 5 7 9 6 4 3 8 ]

18 th: [ 0 1 2 5 7 9 6 4 3 8 ]

19 th: [ 0 1 2 5 7 9 9 4 3 8 ]

20 th: [ 0 1 2 5 7 7 9 4 3 8 ]

21 th: [ 0 1 2 5 6 7 9 4 3 8 ]

22 th: [ 0 1 2 5 6 7 9 9 3 8 ]

23 th: [ 0 1 2 5 6 7 7 9 3 8 ]

24 th: [ 0 1 2 5 6 6 7 9 3 8 ]

25 th: [ 0 1 2 5 5 6 7 9 3 8 ]

26 th: [ 0 1 2 4 5 6 7 9 3 8 ]

27 th: [ 0 1 2 4 5 6 7 9 9 8 ]

28 th: [ 0 1 2 4 5 6 7 7 9 8 ]

29 th: [ 0 1 2 4 5 6 6 7 9 8 ]

30 th: [ 0 1 2 4 5 5 6 7 9 8 ]

31 th: [ 0 1 2 4 4 5 6 7 9 8 ]

32 th: [ 0 1 2 3 4 5 6 7 9 8 ]

33 th: [ 0 1 2 3 4 5 6 7 9 9 ]

34 th: [ 0 1 2 3 4 5 6 7 8 9 ]

#include <stdio.h>

//void simple\_sort(int [],int);

void swap\_data(int [],int,int);

void print\_data(int,int [],int);

//void bubble\_sort(int [],int);

void insertion\_sort(int [],int);

int main(){

int a[] ={9,5,2,7,1,0,6,4,3,8};

int n =10;

print\_data(0,a,n);

printf("\n");

insertion\_sort(a,n);

return 0;

}

void swap\_data(int a[],int i,int j){

int temp;

temp = a[i];

a[i]=a[j];

a[j]=temp;

}

/\*void simple\_sort(int a[],int n){

int i,j;

int times = 1;

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

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

if(a[j]<a[i]){

swap\_data(a,i,j);

}

}

print\_data(times,a,n);

times++;

}

}\*/

void print\_data(int times,int a[],int n){

int i;

printf("%d th: [ ",times);

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

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

}

printf("]\n");

}

/\*void bubble\_sort(int a[],int n){

int i,j;

int times=1;

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

for(j=n-1;j>i;j--){

if(a[j-i]<a[j]){

swap\_data(a,j-1,j);

}

print\_data(times,a,n);

times++;

}

printf("\n");

}

}\*/

void insertion\_sort(int a[],int n){

int i,j;

int w;

int times =1;

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

w=a[i];

j=i-1;

while(j>=0 && w>a[j]){

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

j--;

print\_data(times,a,n);

times++;

}

a[j+1]=w;

print\_data(times,a,n);

times++;

printf("\n");

}

}

実行結果

0 th: [ 9 5 2 7 1 0 6 4 3 8 ]

1 th: [ 9 5 2 7 1 0 6 4 3 8 ]

2 th: [ 9 5 2 7 1 0 6 4 3 8 ]

3 th: [ 9 5 2 2 1 0 6 4 3 8 ]

4 th: [ 9 5 5 2 1 0 6 4 3 8 ]

5 th: [ 9 7 5 2 1 0 6 4 3 8 ]

6 th: [ 9 7 5 2 1 0 6 4 3 8 ]

7 th: [ 9 7 5 2 1 0 6 4 3 8 ]

8 th: [ 9 7 5 2 1 0 0 4 3 8 ]

9 th: [ 9 7 5 2 1 1 0 4 3 8 ]

10 th: [ 9 7 5 2 2 1 0 4 3 8 ]

11 th: [ 9 7 5 5 2 1 0 4 3 8 ]

12 th: [ 9 7 6 5 2 1 0 4 3 8 ]

13 th: [ 9 7 6 5 2 1 0 0 3 8 ]

14 th: [ 9 7 6 5 2 1 1 0 3 8 ]

15 th: [ 9 7 6 5 2 2 1 0 3 8 ]

16 th: [ 9 7 6 5 4 2 1 0 3 8 ]

17 th: [ 9 7 6 5 4 2 1 0 0 8 ]

18 th: [ 9 7 6 5 4 2 1 1 0 8 ]

19 th: [ 9 7 6 5 4 2 2 1 0 8 ]

20 th: [ 9 7 6 5 4 3 2 1 0 8 ]

21 th: [ 9 7 6 5 4 3 2 1 0 0 ]

22 th: [ 9 7 6 5 4 3 2 1 1 0 ]

23 th: [ 9 7 6 5 4 3 2 2 1 0 ]

24 th: [ 9 7 6 5 4 3 3 2 1 0 ]

25 th: [ 9 7 6 5 4 4 3 2 1 0 ]

26 th: [ 9 7 6 5 5 4 3 2 1 0 ]

27 th: [ 9 7 6 6 5 4 3 2 1 0 ]

28 th: [ 9 7 7 6 5 4 3 2 1 0 ]

29 th: [ 9 8 7 6 5 4 3 2 1 0 ]

**3**

#include <stdio.h>

struct point{

double x,y;

};

void simple\_sort(struct point [],int);

void swap\_data(struct point [],int,int);

void print\_data(int,struct point [],int);

int main(){

struct point a[] ={{1,2},{3,4},{5,6},{7,8},{9,0}};

int n =5;

print\_data(0,a,n);

printf("\n");

simple\_sort(a,n);

return 0;

}

void swap\_data(struct point a[],int i,int j){

double tempx,tempy;

tempx = a[i].x;

tempy = a[i].y;

a[i].x=a[j].x;

a[i].y=a[j].y;

a[j].x=tempx;

a[j].y=tempy;

}

void simple\_sort(struct point a[],int n){

int i,j;

int times = 1;

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

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

if(a[j].x<a[i].x){

swap\_data(a,i,j);

}

if(a[j].y<a[i].y){

swap\_data(a,i,j);

}

}

print\_data(times,a,n);

times++;

}

}

void print\_data(int times,struct point a[],int n){

int i;

printf("%d th: [ ",times);

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

printf("%f ",a[i].x);

}

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

printf("%f ",a[i].y);

}

printf("]\n");

}

実行結果

0 th: [ 1.000000 3.000000 5.000000 7.000000 9.000000 2.000000 4.000000 6.000000 8.000000 0.000000 ]

1 th: [ 9.000000 3.000000 5.000000 7.000000 1.000000 0.000000 4.000000 6.000000 8.000000 2.000000 ]

2 th: [ 9.000000 1.000000 5.000000 7.000000 3.000000 0.000000 2.000000 6.000000 8.000000 4.000000 ]

3 th: [ 9.000000 1.000000 3.000000 7.000000 5.000000 0.000000 2.000000 4.000000 8.000000 6.000000 ]

4 th: [ 9.000000 1.000000 3.000000 5.000000 7.000000 0.000000 2.000000 4.000000 6.000000 8.000000 ]

5 th: [ 9.000000 1.000000 3.000000 5.000000 7.000000 0.000000 2.000000 4.000000 6.000000 8.000000 ]