Ian Whittemore c++

Bubble sort

Pseudo code :

Begin

For (I < size - 1)

For (j < size – I – 1)

If(array < array + 1

swap

End

Selection sort

Pseudo code :

Begin

For (I < size – 1)

Min = i

For(d = I + 1; d < size)

If(Arrray[min] < Array[d])

Min = d

else if (min != i)

swap(Array[i], Array[min]

End

Insertion sort

Pseudo code :

Begin

Int num

For(c =1 ; c <= size -1;c++)

Num = c

While(num < 0 && Array[num] < Array[num – 1]

Swap(A[num],A[num – 1]

Num—

End

#include "stdafx.h"

#include <iostream>

using namespace std;

int A[] = { 32 , 56 , 20 , 7 , 12, 2 };

void Question2(int A[], int);

void BubbleSort(int A[], int);

void InsertionSort(int A[], int);

void SelectionSort(int A[], int);

void BubbleInc(int A[], int);

void InsertInc(int A[], int);

void selectInc(int A[], int);

void shellsort(int A[], int );

void Swap(int &X, int &Y);

void Print(int A[], int size);

int main()

{

int size = sizeof(A) / sizeof(int);

cout << "Question 2 " << endl;

Question2(A, size);

Print(A, size);

cout << "Question 3.1" << endl;

BubbleSort(A, size);

Print(A, size);

cout << "Question 3.2" << endl;

InsertionSort(A, size);

Print(A, size);

cout << "Question 3.3" << endl;

SelectionSort(A, size);

Print(A, size);

cout << "Question 3.4" << endl;

BubbleInc(A, size);

Print(A, size);

cout << "Question 3.5" << endl;

InsertInc(A, size);

Print(A, size);

cout << "Question 3.6" << endl;

selectInc(A, size);

Print(A, size);

cout << "Shellsort " << endl;

Print(A, size);

shellsort(A, size);

Print(A, size);

return 0;

}

void Swap(int &X, int &Y)

{

int temp = X;

X = Y;

Y = temp;

}

void Print(int A[], int size)

{

cout << " = ";

for (int i = 0; i < size; i++)

{

cout << A[i] << " " << endl;

}

}

void Question2(int A[], int size)

{

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

{

for (int d = i - size; d < size - 1 - i; d++)

{

if (A[d] > A[d + 1])

{

Swap(A[d], A[d + 1]);

}

}

}

}

void BubbleSort(int A[], int size)

{

int n = 0;

int count = 0;

for (int c = 0; c < size - 1; c++)

{

for (int d = 0; d < size - c - 1; d++)

{

count++;

if (A[d] > A[d + 1])

{

Swap([d], A[d + 1]);

n++;

cout << "Checks : " << count << endl;

cout << "swaps : " << n << endl;

}

}

}

}

void InsertionSort(int A[], int size)

{

int d;

int count = 0;

int n = 0;

for (int c = 1; c <= size - 1; c++)

{

d = c;

while (d > 0)

{

count++;

if (A[d] < A[d - 1])

{

Swap( A[d] < A[d - 1]);

d--

n++;

cout << "Checks : " << count << endl;

cout << "swaps : " << n << endl;

}

}

}

}

void SelectionSort(int A[], int size)

{

int min, c, d;

int count =0;

int n = 0;

for (c = 0; c < size - 1; c++)

{

min = c;

for (d = c + 1; d < size; d++)

{

if (A[min] > A[d])

{

min = d;

count++;

}

if (min != c)

{

Swap(A[c], A[min]);

n++;

cout << "Checks : " << count << endl;

cout << "swaps : " << n << endl;

}

}

}

}

void BubbleInc(int A[], int size)

{

for (int c = 0; c < size - 1; c++)

{

for (int d = 0; d < size - c - 1; d++)

{

if (A[d] < A[d + 1])

{

Swap([d], A[d + 1]);

}

}

}

}

void InsertInc(int A[], int size)

{

int d;

for (int c = 1; c <= size - 1; c++)

{

d = c;

while (d > 0)

{

if (A[d] > A[d - 1])

{

Swap(A[d] < A[d - 1]);

d--

}

}

}

}

void selectInc(int A[], int size)

{

int min, c, d;

for (c = 0; c < size - 1; c++)

{

min = c;

for (d = c + 1; d < size; d++)

{

if (A[min] < A[d])

{

min = d;

}

if (min != c)

{

Swap(A[c], A[min]);

}

}

}

}

void shellsort(int A[], int size)

{

int temp;

int count = 0;

int n = 0;

for (int i = size / 2; i > 0; i = i / 2)

{

for (int j = i; j < size; j++)

{

for (int k = j - i; k >= 0; k = k - i)

{

count++;

if (A[k + j] >= A[k])

{

break;

}

else

{

n++;

temp = A[k];

A[k] = A[k + i];

A[k + i] = temp;

}

}

}

cout << "Checks : " << count << endl;

cout << "swaps : " << n << endl;

}

}