|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | i |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 2 | 5 | 4 | 6 | 2 | 7 | 5 |
|  |  |  |  |  |  |  |  |  |  |  |
|  | kk | 2 |  | 1 | m |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

// сортировка массива методом "ЧЕРЕЗ ОТБОР"

void srt1(int kl, int \*ms)

{

int i, j, m, kk,s=0;

for (i = 0; i<kl - 1; ++i)

{

m = i; // индекс элемента для упорядочивания

kk = ms[i]; // значение элемента

for (j = i + 1; j<kl; ++j) // перебор последующих эл-тов справа от i-го

if (ms[j]<kk) // найден меньший

{

m = j; // запоминаем его индекс и значение

kk = ms[j];

s++;

}

ms[m] = ms[i]; // замена выбранного и меньшего

ms[i] = kk;

}

}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  | i |  |  |  |  |  |  |  |  |
|  | 2 | 2 | 3 | 4 | 5 | 1 | 6 | 2 | 7 | 5 |
|  | J |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 2 | kk |  |  |  |  |  |

// сортировка массива методом " ВСТАВКИ "

void srt2(int kl, int ms[])

{

int i, j, kk;

for (i = 1; i<kl; ++i) // индекс элемента для упорядочивания

{

j = i - 1; // индекс предыдущего элемента

kk = ms[i]; // значение предыдущего элемента

while (j >= 0 && kk<ms[j])

ms[j-- + 1] = ms[j]; // сдвиг всех элементов направо

ms[j + 1] = kk; // запись в освободившийся или в тот же элемент

}

}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | **i** | **🡪** |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 4 | 2 | 3 | 2 | 5 | 5 | 6 | 7 |
|  |  |  |  |  |  |  |  | **🡨** | **j-1** | **j** |
|  |  |  |  |  |  |  |  |  |  |  |

// сортировка чисел методом "ПУЗЫРЬКА"

void srt3(int kl, int \*ms)

{

int i, j, m;

for (i = 0; i<kl - 1; ++i) // цикл выбора эл-тов до последнего к упорядочиванию

for (j = kl - 1; j>i; --j) // цикл перебора эл-тов с конечного до i-го

{

if (ms[j - 1]>ms[j]) // если (j-1)-й > j-го элемента, то

{ // выполныется их перестановка

m = ms[j - 1];

ms[j - 1] = ms[j];

ms[j] = m;

}

}

}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  | **🡪** | **с** |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 2 | 4 | 2 | 5 | 6 | 6 | 7 |
|  |  |  |  |  |  |  | **d** |  | **🡨** |  |
|  |  |  |  | **b** | **6** |  |  |  |  |  |

void shaker(int kl, int \*ms)

{

int i, a, b, c, d;

c = 1;

b = kl - 1; //номер элемента на котором остановка

d = kl - 1; //номер стартового элемента для сортировки справа налево

do

{

for (i = d; i >= c; --i) // цикл попарного сравнения элементов

{ // справа налево по массиву

if (ms[i - 1]>ms[i]) // выполняется условие перестановки

{ // (i-1)-ого и i-го элемента

a = ms[i - 1];

ms[i - 1] = ms[i];

ms[i] = a;

b = i; // запоминаем крайний слева упорядоченный элемент

}

}

c = b + 1; // номер стартового элемента для сортировки слева направо

for (i = c; i <= d; ++i) // аналогично предыдущему циклу, но движение

{ // слева направо по массиву

if (ms[i - 1]>ms[i])

{

a = ms[i - 1];

ms[i - 1] = ms[i];

ms[i] = a;

b = i; // крайний слева упорядоченный элемент

}

}

d = b - 1;

} while (c <= d);

}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| kl | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **dp** | **i** |
|  | 4 | 2 | 3 | 2 | 5 | 1 | 6 | 2 | 7 | 5 |  | 5 | 5…kl |
|  | 1 | 2 | 2 | 2 | 5 | 4 | 6 | 3 | 7 | 5 |  |  |  |
|  | 1 | 2 | 2 | 2 | 5 | 4 | 6 | 3 | 7 | 5 |  | 2 | 2…kl |
|  | 1 | 2 | 2 | 2 | 5 | 4 | 6 | 3 | 7 | 5 |  |  |  |
|  | 1 | 2 | 2 | 2 | 5 | 3 | 6 | 4 | 7 | 5 |  |  |  |
|  | 1 | 2 | 2 | 2 | 5 | 3 | 6 | 4 | 7 | 5 |  | 1 | 1…kl |
|  | 1 | 2 | 2 | 2 | 3 | 5 | 6 | 4 | 7 | 5 |  |  |  |
|  | 1 | 2 | 2 | 2 | 3 | 5 | 4 | 6 | 7 | 5 |  |  |  |
|  | 1 | 2 | 2 | 2 | 3 | 5 | 4 | 6 | 5 | 7 |  |  |  |
|  | 1 | 2 | 2 | 2 | 3 | 5 | 4 | 5 | 6 | 7 |  |  |  |
|  | 1 | 2 | 2 | 2 | 3 | 4 | 5 | 5 | 6 | 7 |  |  |  |

// "Шелл" сортировка чисел

void shell(int kl, int ms[])

{

int i, j, dp, a;

for (dp = kl / 2; dp>0; dp /= 2)

for (i = dp; i<kl; i++)

for (j = i - dp; j >= 0 && ms[j]>ms[j + dp]; j -= dp)

{

a = ms[j];

ms[j] = ms[j + dp];

ms[j + dp] = a;

}

}