// сортировка пузырьком

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

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

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

swap (a[j], a[j + 1]);

}

}

// сортировка расческой

int gap = n;

bool flag = true;

while ((gap > 1) || flag){

gap = (gap \* 10) / 13;

if (gap < 1)

gap = 1;

flag = false;

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

count\_1++;

if(a[i] > a[i+gap]){

count\_2++;

swap (a[i], a[i + gap]);

flag = true;

}

}

}

// сортировка выбором

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

int mini = i;

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

if (a[j] < a[mini])

mini = j;

}

swap(a[mini], a[i]);

}

// шейкерная сортировка

int j, k = size - 1, lb = 0, rb = size - 1;

while (lb < rb){

for (j = rb; j > 0; j--){

if (a[j - 1] > a[j])

swap (a[j - 1], a[j]);

lb = k + 1;

}

for (j = 1; j <=rb; j++){

if (a[j - 1] > a[j]){

swap (a[j - 1], a[j]);

k = j;

}

}

rb = k - 1;

}

// сортировка подсчетом

int \*count = new int[n];

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

count[i] = 0;

}

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

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

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

count[j]++;

else

count[i]++;

}

}

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

r[count[i]] = a[i];

}

// сортировка вставками

int x;

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

x = a[i];

for (int j = i - 1; j >= 0 && a[j] > x; j--)

a[j + 1] = a[j]; //сдвиг элементов вправо

a[j + 1] = x //вставка элемента

}

// сортировка Шелла

int h, k;

h = n / 2;

while (h > 0){

k = 1;

while (k){

k = 0;

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

if (a[i] > a[i + h]){

swap (a[i], a[i + h]);

k = 1;

}

}

}

h /= 2;

}