|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  Санкт-Петербургский политехнический университет Петра Великого  Институт компьютерных наук и технологий  Высшая школа киберфизических систем и управления | |
| ОТЧЁТ  по дисциплине «Теория и технология программирования»  по лабораторной работе №1 | |
| **Выполнил**: |  |
| студент гр. 3530902/10003 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сурьянинова В. В. |
|  | дата, подпись |
|  |  |
| **Проверил:**  ассистент |  |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Журавская А. |
|  | дата, подпись |
|  | |

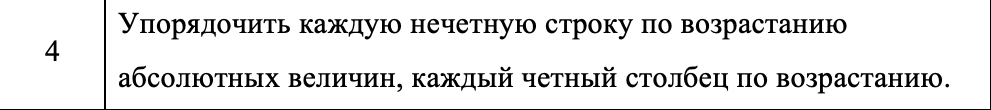
1. Задание

На рисунке 1 представлено задание, на рисунке 2 представлен мой вариант.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1



Рисунок

1. Текст программы

include <iostream>

#include <ctime>

#include <math.h>

#include <string>

**using** **namespace** std;

**void** swap(**int**& a, **int**& b){

**int** t = a;

a = b;

b = t;

}

**int** enter(string& k, **char** j){

**bool** f = **true**;

**int** count, t;

**while** (f == **true**) {

cout << "Enter the " << j << ": ";

k = "";

cin >> k;

count = 0;

**for**(**int** i = 0; i < k.length(); i++){

**if** ((k[i] >= '0')&&(k[i] <= '9')){

count++;

}

}

**if** (count == k.length()){

f = **false**;

}

}

t = 0;

**for**(**int** i = 0; i < k.length(); i++){

t = 10 \* t + k[i] - 48;

}

**return** t;

}

**void** bubble(**int** a[], **int** n, **int**& k, **int**& s){

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

**int** i = 0;

**while**(i < n - l - 1){

s++;

**if** (a[i] > a[i + 1]){

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

k++;

}

i++;

}

}

}

**void** choice(**int** a[], **int** n, **int**& k, **int**& s){

**int** min = a[0], min\_i = 0;

**int** i = 0;

**while** (i < n){

min = a[i];

min\_i = i;

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

s++;

**if** (min > a[j]){

min = a[j];

min\_i = j;

}

}

s++;

**if**(a[min\_i] != a[i]){

swap(a[min\_i], a[i]);

k++;

}

i++;

}

}

**void** inset(**int** a[], **int** n, **int**& k, **int**& s){

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

**for**(**int** j = i; (j > 0); j--){

s++;

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

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

k++;

}

}

}

}

**void** shell(**int** a[], **int** n, **int**& k, **int**& s){

**int** l, j;

**for**(**int** t = n/2; t > 0; t = t / 2){

**for**(**int** i = t; i < n; i++){

l = a[i];

**for**(j = i; j >= t; j -= t){

s++;

**if** (l < a[j - t]){

a[j] = a[j - t];

k++;

}

**else** **break**;

}

a[j] = l;

}

}

}

**void** quicksort(**int** \*a, **int** n, **int**& k, **int**& s){

**int** i = 0;

**int** j = n - 1;

**int** mid = a[n / 2];

**do** {

s++;

**while**(a[i] < mid) {

i++;

}

s++;

**while**(a[j] > mid) {

j--;

}

**if** (i <= j) {

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

**if** (a[i] != a[j]) k++;

i++;

j--;

}

} **while** (i <= j);

**if**(j > 0) {

quicksort(a, j + 1, k, s);

}

**if** (i < n) {

quicksort(&a[i], n - i, k, s);

}

}

**int** main() {

srand(time(0));

**int** n, m, count = 0, d\_count = 0, s = 0, d\_s = 0;

string q;

**int** res[2][5];

n = enter(q, 'N');

m = enter(q, 'M');

**int** a[n][m], b[n][m], c[m], d[n], c1[m], d1[n];

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

**for**(**int** j = 0; j < m; j++){

a[i][j] = rand() % 50;

}

}

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

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", a[i][j]);

}

cout << "\n";

}

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

**for**(**int** j = 0; j < m; j++){

b[i][j] = a[i][j];

}

}

cout << "BUBBLE\n";

**while** (d\_count <= count) {

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

**if** (i % 2 != 0){

**for**(**int** j = 0; j < m; j++){

c[j] = b[i][j];

}

bubble(c, m, count, s);

**for**(**int** j = 0; j < m; j++){

b[i][j] = c[j];

}

}

}

**for**(**int** j = 0; j < m; j++){

**if** (j % 2 == 0){

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

d[i] = b[i][j];

}

bubble(d, n, count, s);

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

b[i][j] = d[i];

}

}

}

**if** (d\_count == count){

d\_count += 1;

s = d\_s;

}

**else** {

d\_count = count;

d\_s = s;

}

}

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

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", b[i][j]);

}

cout << "\n";

}

res[0][0] = count;

res[1][0] = s;

count = 0, d\_count = 0, s = 0, d\_s = 0;

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

**for**(**int** j = 0; j < m; j++){

b[i][j] = a[i][j];

}

}

cout << "CHOICE\n";

**while** (d\_count <= count) {

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

**if** (i % 2 != 0){

**for**(**int** j = 0; j < m; j++){

c[j] = b[i][j];

}

choice(c, m, count, s);

**for**(**int** j = 0; j < m; j++){

b[i][j] = c[j];

}

}

}

**for**(**int** j = 0; j < m; j++){

**if** (j % 2 == 0){

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

d[i] = b[i][j];

}

choice(d, n, count, s);

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

b[i][j] = d[i];

}

}

}

**if** (d\_count == count){

d\_count += 1;

s = d\_s;

}

**else** {

d\_count = count;

d\_s = s;

}

}

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

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", b[i][j]);

}

cout << "\n";

}

res[0][1] = count;

res[1][1] = s;

count = 0, d\_count = 0, s = 0, d\_s = 0;

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

**for**(**int** j = 0; j < m; j++){

b[i][j] = a[i][j];

}

}

cout << "INSET\n";

**while** (d\_count <= count) {

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

**if** (i % 2 != 0){

**for**(**int** j = 0; j < m; j++){

c[j] = b[i][j];

}

inset(c, m, count, s);

**for**(**int** j = 0; j < m; j++){

b[i][j] = c[j];

}

}

}

**for**(**int** j = 0; j < m; j++){

**if** (j % 2 == 0){

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

d[i] = b[i][j];

}

inset(d, n, count, s);

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

b[i][j] = d[i];

}

}

}

**if** (d\_count == count){

d\_count += 1;

s = d\_s;

}

**else** {

d\_count = count;

d\_s = s;

}

}

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

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", b[i][j]);

}

cout << "\n";

}

res[0][2] = count;

res[1][2] = s;

count = 0, d\_count = 0, s = 0, d\_s = 0;

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

**for**(**int** j = 0; j < m; j++){

b[i][j] = a[i][j];

}

}

cout << "SHELL\n";

**while** (d\_count <= count) {

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

**if** (i % 2 != 0){

**for**(**int** j = 0; j < m; j++){

c[j] = b[i][j];

}

shell(c, m, count, s);

**for**(**int** j = 0; j < m; j++){

b[i][j] = c[j];

}

}

}

**for**(**int** j = 0; j < m; j++){

**if** (j % 2 == 0){

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

d[i] = b[i][j];

}

shell(d, n, count, s);

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

b[i][j] = d[i];

}

}

}

**if** (d\_count == count){

d\_count += 1;

s = d\_s;

}

**else** {

d\_count = count;

d\_s = s;

}

}

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

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", b[i][j]);

}

cout << "\n";

}

res[0][3] = count;

res[1][3] = s;

count = 0, d\_count = 0, s = 0, d\_s = 0;

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

**for**(**int** j = 0; j < m; j++){

b[i][j] = a[i][j];

}

}

cout << "QUICK\n";

**while** (d\_count <= count) {

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

**if** (i % 2 != 0){

**for**(**int** j = 0; j < m; j++){

c[j] = b[i][j];

}

quicksort(c, m, count, s);

**for**(**int** j = 0; j < m; j++){

b[i][j] = c[j];

}

}

}

**for**(**int** j = 0; j < m; j++){

**if** (j % 2 == 0){

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

d[i] = b[i][j];

}

quicksort(d, n, count, s);

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

b[i][j] = d[i];

}

}

}

**if** (d\_count == count){

d\_count += 1;

s = d\_s;

}

**else** {

d\_count = count;

d\_s = s;

}

}

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

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", b[i][j]);

}

cout << "\n";

}

res[0][4] = count;

res[1][4] = s;

printf("||-----------||-----------||-----------||-----------||-----------||-----------||\n");

printf("|| || bubble || choice || inset || shell || quick ||\n");

printf("||-----------||-----------||-----------||-----------||-----------||-----------||\n");

printf("|| reshuffle ||%11i||%11i||%11i||%11i||%11i||\n", res[0][0], res[0][1], res[0][2], res[0][3], res[0][4]);

printf("||-----------||-----------||-----------||-----------||-----------||-----------||\n");

printf("||comparisons||%11i||%11i||%11i||%11i||%11i||\n", res[1][0], res[1][1], res[1][2], res[1][3], res[1][4]);

printf("||-----------||-----------||-----------||-----------||-----------||-----------||\n");

}

1. Входные и выходные данные

Входные данные: две целочисленные переменные - размеры матрицы.

Выходные данные:

1. Исходная матрица, заполненная случайными числами,
2. Матрица, отсортированная пятью разными способами сортировки,
3. Таблица, содержащая информацию о количестве перестановок и сравнений в каждом виде сортировок.
4. Пример работы программы

На рисунке 3 и рисунке 4 представлены примеры моей выполненной программы.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5

1. Вывод

В результате проделанной мною работы я написала программу, которая сортирует матрицу по столбцам и строкам разными способами. Во время выполнения данной лабораторной работы я ознакомилась с основами написания программ на C++, а также смогла понять различие в эффективности различных сортировок, что поможет мне в написании более сложных программ в будущем.