

**Московский государственный технический
университет им. Н. Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управление»

Курс «Основы программирования»

Отчет по лабораторной работе №4
«Сортировка одномерного числового массива»

Выполнил:

Студент группы ИУ5-11Б

Алехин Сергей

Подпись и дата:

Проверил:

Преподаватель каф. ИУ5

Правдина Анна Дмитриевна

Подпись и дата:

Москва, 2018 г.

Задание

Отсортировать числовой массив методом выбора максимального (минимального) элемента и методом пузырькового всплытия. По окончании сортировки вывести отсортированный массив и количество сделанных сравнений и перестановок элементов.

Сравнить быстродействие алгоритмов, которое определяется числом сравнений и перестановок, для исходного не отсортированного массива и для исходного массива, отсортированного в прямом и обратном порядке.

Исследовать зависимость быстродействия от размера массива. Возможность изменения длины массива реализуйте с помощью динамического массива, а для его инициализации используйте датчик случайных чисел (см. Приложение 1). Результаты исследования выведите в виде отформатированной таблицы.

Разработка алгоритма

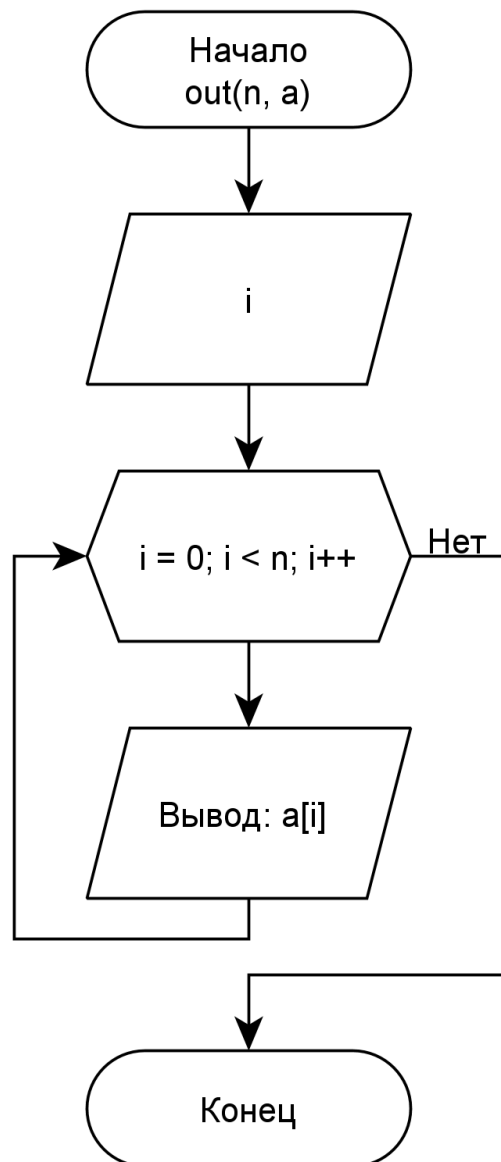
Входные переменные:

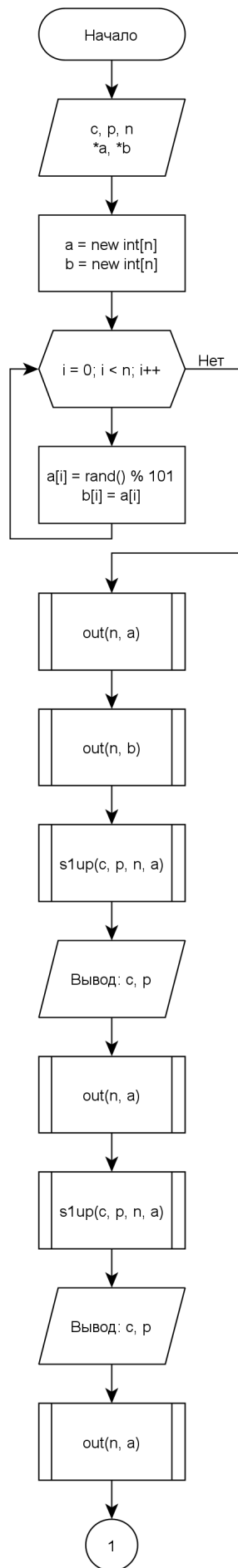
- 1) int c – количество сочетаний;
- 2) int p – количество перестановок;
- 3) int n – количество элементов в динамическом массиве;
- 4) int *a – динамический массив a;
- 5) int *b – динамический массив b;
- 6) int i – переменная цикла;

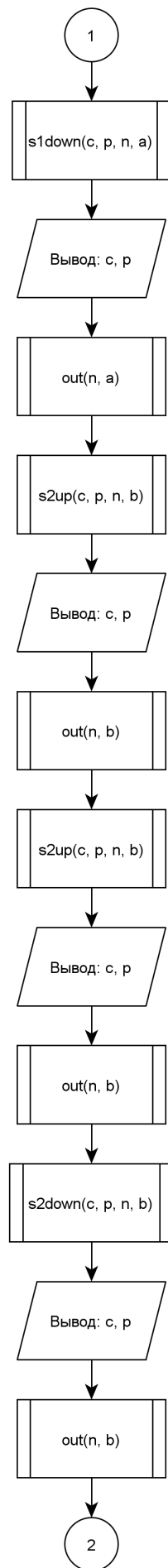
Функции:

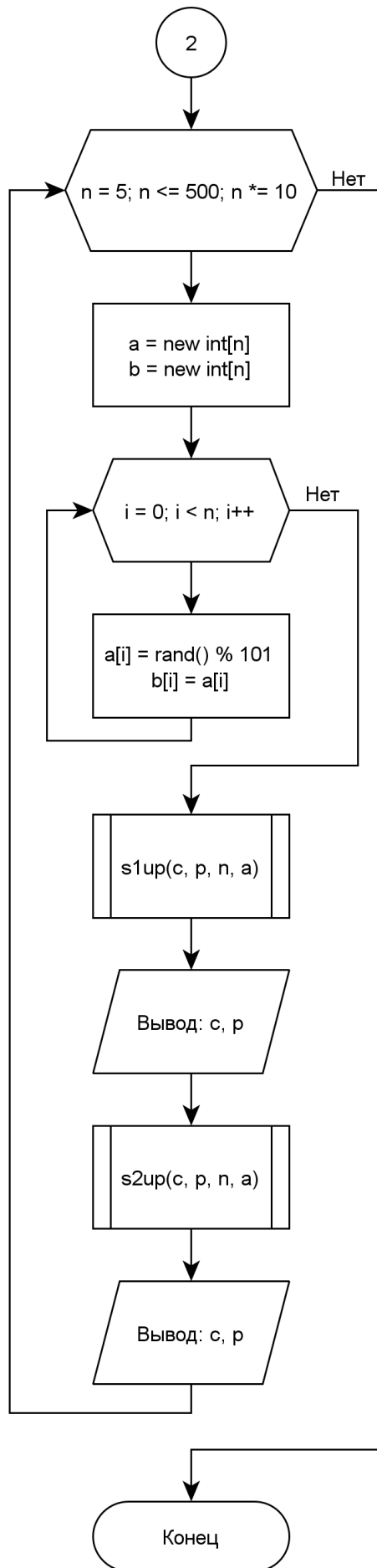
- 1) void s1up – метод пузырька по возрастанию;
 - a. *Входные переменные:*
 - i. int &c – ссылочная переменная на c, количество сочетаний;
 - ii. int &p – ссылочная переменная на p, количество перестановок;
 - iii. int n – длина динамического массива;
 - iv. int *a – динамический массив a;
 - b. *Локальные переменные:*
 - i. int k – количество перестановок за 1 цикл;
 - ii. int i – переменная внешнего цикла;
 - iii. int j – переменная внутреннего цикла;
- 2) void s1down – метод пузырька по убыванию;
 - a. *Входные переменные:*
 - i. int &c – ссылочная переменная на c, количество сочетаний;
 - ii. int &p – ссылочная переменная на p, количество перестановок;
 - iii. int n – длина динамического массива;
 - iv. int *a – динамический массив a;
 - b. *Локальные переменные:*
 - i. int k – количество перестановок за 1 цикл;
 - ii. int i – переменная внешнего цикла;
 - iii. int j – переменная внутреннего цикла;
- 3) void s2up – метод min/max по возрастанию;
 - a. *Входные переменные:*
 - i. int &c – ссылочная переменная на c, количество сочетаний;
 - ii. int &p – ссылочная переменная на p, количество перестановок;
 - iii. int n – длина динамического массива;
 - iv. int *b – динамический массив b;
 - b. *Локальные переменные:*
 - i. int imin – индекс найденного максимального элемента;
 - ii. int i – переменная внешнего цикла;
 - iii. int j – переменная внутреннего цикла;

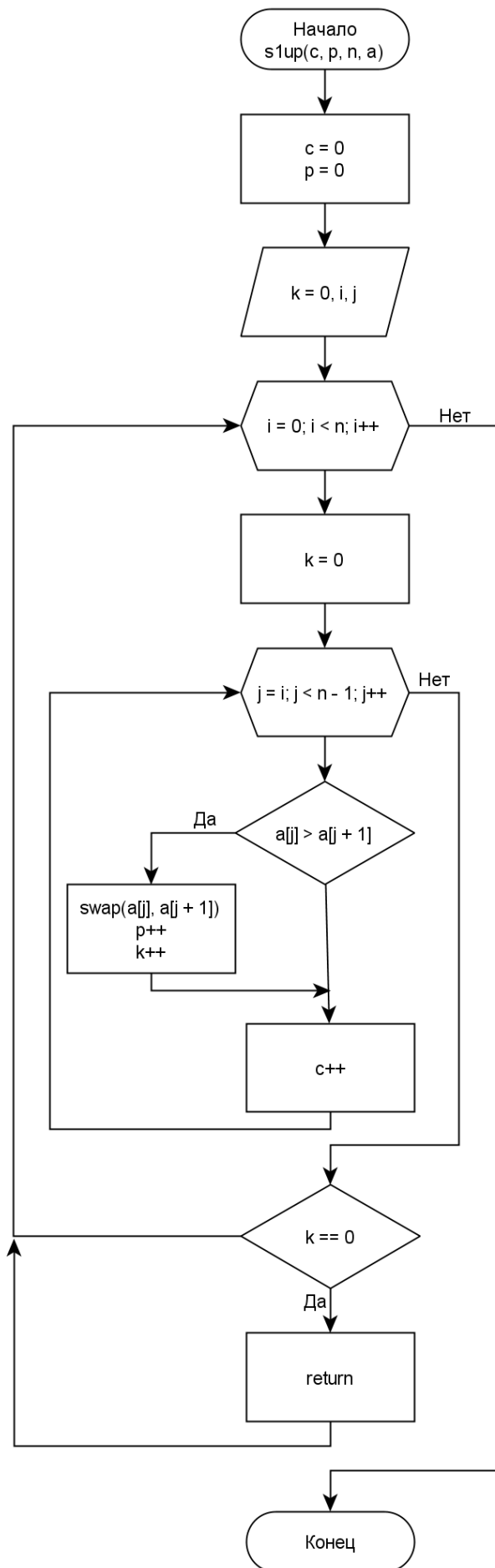
- 4) void s2down – метод min/max по убыванию;
- a. *Входные переменные:*
 - i. int &c – ссылочная переменная на c, количество сочетаний;
 - ii. int &p – ссылочная переменная на p, количество перестановок;
 - iii. int n – длина динамического массива;
 - iv. int *b – динамический массив b;
 - b. *Локальные переменные:*
 - i. int imin – индекс найденного максимального элемента;
 - ii. int i – переменная внешнего цикла;
 - iii. int j – переменная внутреннего цикла;
- 5) void out – вывод массива на экран;
- a. *Входные переменные:*
 - i. int n – длина динамического массива;
 - ii. int *a – динамический массив a или b;
 - b. *Локальные переменные:*
 - i. int i – переменная цикла;

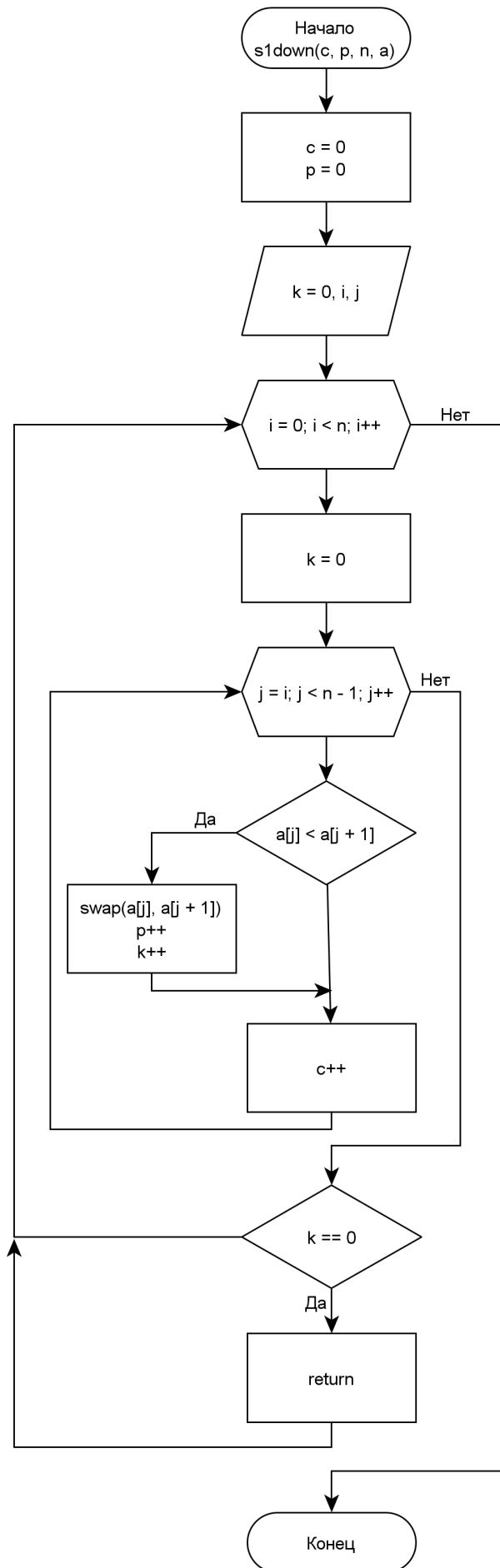


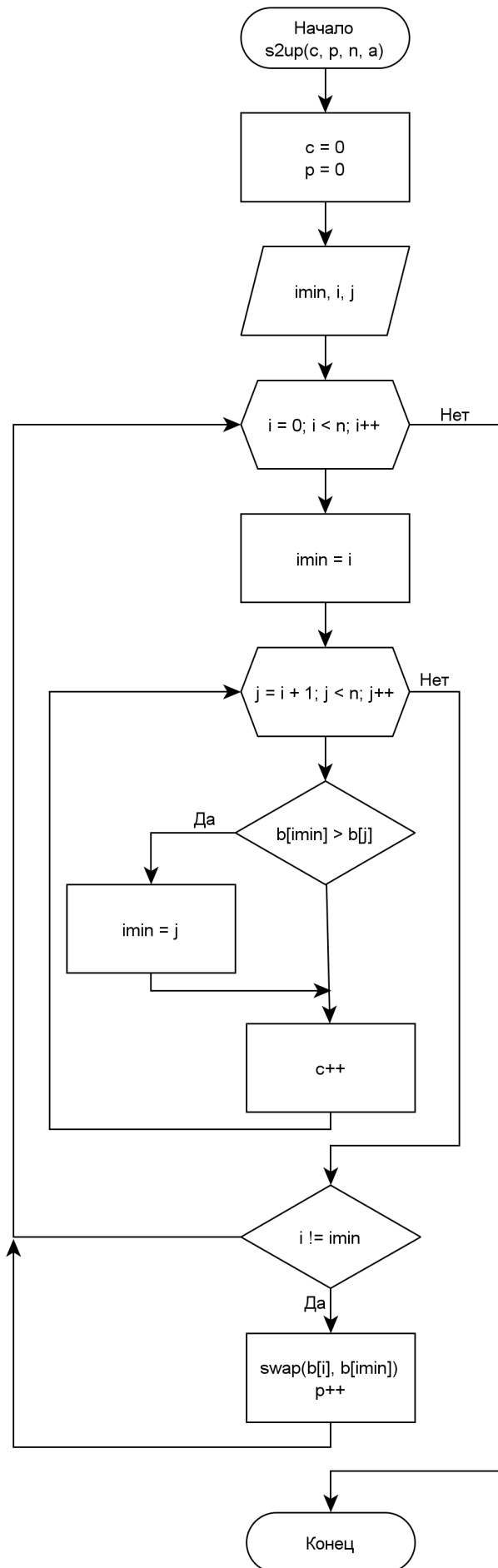


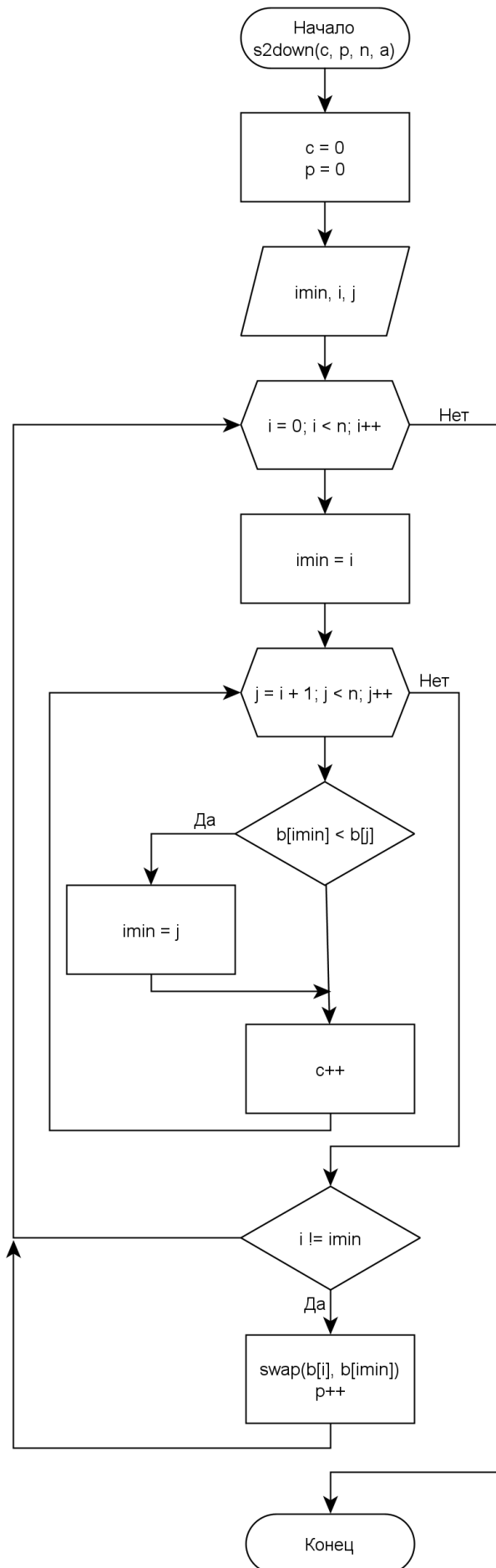












Текст программы

```
#include <iostream>
#include <iomanip>

using namespace std;

void s1up(int &, int &, int, int *);
void s1down(int &, int &, int, int *);
void s2up(int &, int &, int, int *);
void s2down(int &, int &, int, int *);
void out(int, int *);

int main()
{
    srand(unsigned(time(0)));
    setlocale(0, "RUSSIAN");
    int c = 0, p = 0, n = 4;
    int *a, *b;

    a = new int[n];
    b = new int[n];

    for (int i = 0; i < n; i++)
    {
        a[i] = rand() % 101;
        b[i] = a[i];
    }

    out(n, a);
    out(n, b);

    s1up(c, p, n, a);
    cout << "Метод пузырька, по возрастанию " << "сравнений = " << c << "
перестановок = " << p << endl << endl;
    out(n, a);

    s1up(c, p, n, a);
    cout << "Метод пузырька, по возрастанию " << "сравнений = " << c << "
перестановок = " << p << endl << endl;
    out(n, a);

    s1down(c, p, n, a);
    cout << "Метод пузырька, по убыванию " << "сравнений = " << c << "
перестановок = " << p << endl << endl;
    out(n, a);

    s2up(c, p, n, b);
    cout << "Метод min/max, по возрастанию " << "сравнений = " << c << "
перестановок = " << p << endl << endl;
    out(n, b);
}
```

```

        s2up(c, p, n, b);
        cout << "Метод min/max, по возрастанию " << "сравнений = " << c << "
перестановок = " << p << endl << endl;
        out(n, b);

        s2down(c, p, n, b);
        cout << "Метод min/max, по убыванию " << "сравнений = " << c << "
перестановок = " << p << endl << endl;
        out(n, b);

        delete a;
        delete b;

        cout << "Оценка методов сортировки (По возрастанию)" << endl;
        cout << setw(16) << "1 метод" << setw(21) << "2 метод" << endl;

        for (n = 5; n <= 500; n *= 10)
        {
            a = new int[n];
            b = new int[n];

            for (int i = 0; i < n; i++)
            {
                a[i] = rand() % 101;
                b[i] = a[i];
            }

            s1up(c, p, n, a);
            cout << setw(3) << left << n << " c = " << setw(6) << left << c << " p = " <<
            setw(5) << left << p;

            s2up(c, p, n, b);
            cout << " c = " << setw(6) << left << c << " p = " << setw(3) << left << p <<
            endl;

            delete a;
            delete b;
        }
        return 0;
    }

    void s1up(int &c, int &p, int n, int *a)
    {
        c = 0; p = 0;
        int k = 0;
        for (int i = 0; i < n; i++)
        {
            k = 0;
            for (int j = i; j < n - 1; j++)
            {
                if (a[j] > a[j + 1])
                {

```

```

        swap(a[j], a[j + 1]);
        p++;
        k++;
    }
    c++;
}
if (k == 0) return;
}
}

```

```

void s1down(int &c, int &p, int n, int *a)
{
    c = 0; p = 0;
    int k = 0;
    for (int i = 0; i < n - 1; i++)
    {
        k = 0;
        for (int j = 0; j < n - i - 1; j++)
        {
            if (a[j] < a[j + 1])
            {
                swap(a[j], a[j + 1]);
                p++;
                k++;
            }
            c++;
        }
        if (k == 0) return;
    }
}

```

```

void s2up(int &c, int &p, int n, int *b)
{
    c = 0; p = 0;
    int imin;
    for (int i = 0; i < n; i++)
    {
        imin = i;
        for (int j = i + 1; j < n; j++)
        {
            if (b[imin] > b[j])
                imin = j;
            c++;
        }
        if (i != imin)
        {
            swap(b[i], b[imin]);
            p++;
        }
    }
}

```

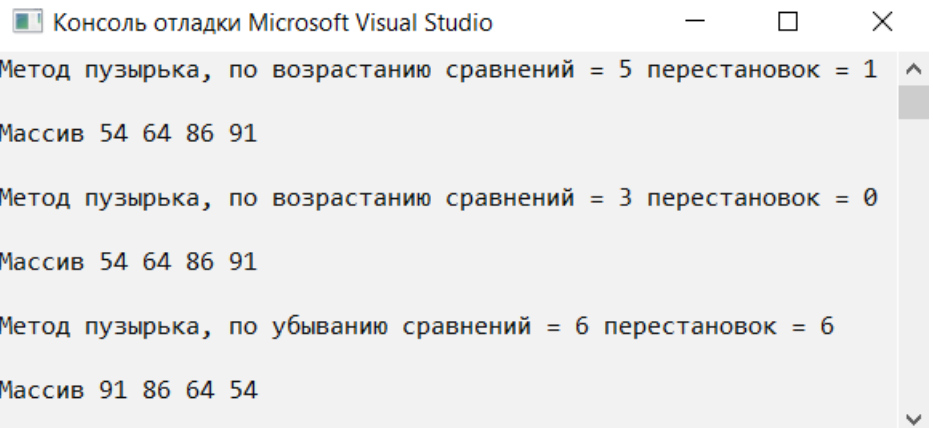
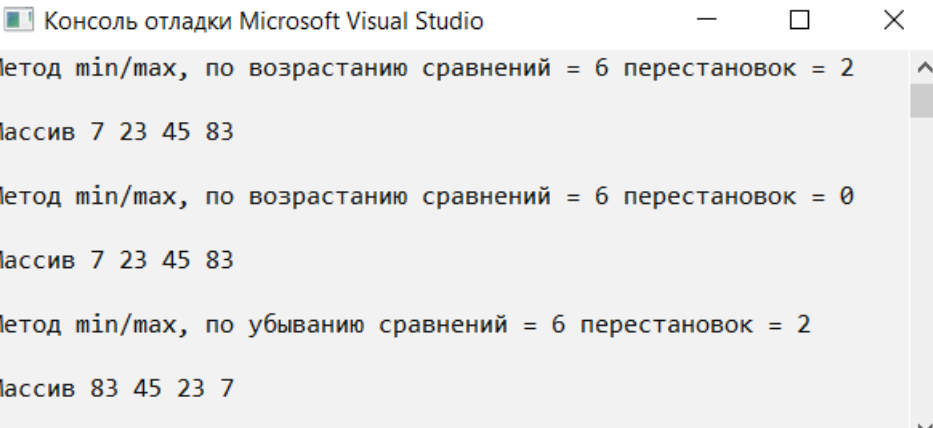
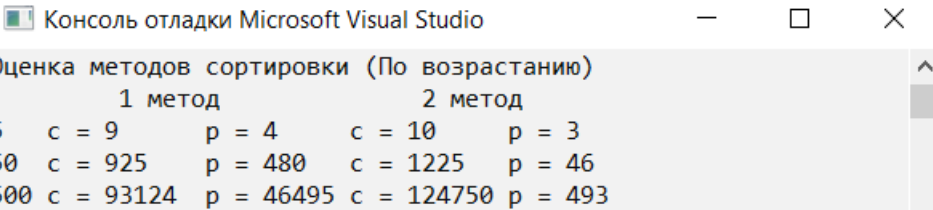
```

void s2down(int &c, int &p, int n, int *b)
{
    c = 0; p = 0;
    int imin;
    for (int i = 0; i < n; i++)
    {
        imin = i;
        for (int j = i + 1; j < n; j++)
        {
            if (b[imin] < b[j])
                imin = j;
            c++;
        }
        if (i != imin)
        {
            swap(b[i], b[imin]);
            p++;
        }
    }
}

void out(int n, int *a)
{
    cout << "Массив ";
    for (int i = 0; i < n; i++)
        cout << a[i] << ' ';
    cout << endl << endl;
}

```

Анализ результатов

№	Входные данные	Полученный результат
1	Сортировка методом пузырька	 <pre> Консоль отладки Microsoft Visual Studio Метод пузырька, по возрастанию сравнений = 5 перестановок = 1 Массив 54 64 86 91 Метод пузырька, по возрастанию сравнений = 3 перестановок = 0 Массив 54 64 86 91 Метод пузырька, по убыванию сравнений = 6 перестановок = 6 Массив 91 86 64 54 </pre>
2	Сортировка методом min/max	 <pre> Консоль отладки Microsoft Visual Studio Метод min/max, по возрастанию сравнений = 6 перестановок = 2 Массив 7 23 45 83 Метод min/max, по возрастанию сравнений = 6 перестановок = 0 Массив 7 23 45 83 Метод min/max, по убыванию сравнений = 6 перестановок = 2 Массив 83 45 23 7 </pre>
3	Сравнение методов сортировки	 <pre> Консоль отладки Microsoft Visual Studio Оценка методов сортировки (По возрастанию) 1 метод 2 метод 5 c = 9 p = 4 c = 10 p = 3 50 c = 925 p = 480 c = 1225 p = 46 500 c = 93124 p = 46495 c = 124750 p = 493 </pre>

Вывод: Мы изучили 2 метода сортировки, и выяснили что при большом количестве значений выгодно использовать метод сортировки min/max, потому что он совершает намного меньше перестановок.