



Методы сортировки

ОТЧЕТ

по лабораторной работе № 5

по дисциплине

Технология программирования

РУКОВОДИТЕЛЬ:

(подпись)

Капранов С. Н.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Чеботова М. С.
(фамилия, и.,о.)

18-ИСТ-4
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Вариант 20

Задание:

Отсортировать последовательность так, чтобы вначале шли отсортированные четные элементы, а затем отсортированные нечетные.

Листинг программы:

```
//Сортировка вставками

#include <iostream>
#include <vector>
#include <string>

using namespace std;

int n = 0; //кол-во элементов
vector <int> arr; //массив

bool even(int i) {
    return (i % 2 == 0);
}

//функция вывода в консоль массив
void print() {
    cout << "Вывод массива...\n";
    for (int i : arr)
    {
        cout << i << ' ';
    }
}

//функция считывания Double (используется для длины и высоты прямоугольников)
void inputInt(string str, int& d)
{
    while (true) //цикл в котором запрашиваем ввод у пользователя
    {
        cout << str;
        cin >> d;
        if (d > 0)
        {
            break;
        }
        if (cin.fail())
        {
            cin.clear();
            std::cin.ignore(32767, '\n');
        }
        cout << "Значение не допустимо!\n";
    }
}
```

```

int main()
{
    setlocale(LC_ALL, "Russian");

    inputInt("Введите количество элементов:\n", n);
    cout << "Введите элементы:\n";
    int x = 0;
    for (int i = 0; i < n; ++i) {
        inputInt("Ввод " + to_string(i + 1) + " элемента\n", x);
        arr.push_back(x); //добавляем его в массив
    }
    cout << "Массив успешно считан.\n";

    /*принцип сортировки такой: смотрим все пары элементов.
    i-ый элемент всегда находится перед j-ым. Поэтому мы смотрим на j-ый
элемент.
    Если он должен стоять перед i-ым тогда меняем их местами.
    */


    //функция even возвращает true если аргумент четный, иначе false;
    for (int i = 0; i < n - 1; ++i)
    {
        for (int j = i + 1; j < n; ++j)
        {
            if (even(arr[j]))
            { /*если второй элемент четный. Тогда нам нужно его менять с
первым:
                                                    1)если первый нечетный
                                                    2)если первый тоже четный и он больше
второго
                                                    */
                if (!even(arr[i]) || arr[i] > arr[j])
                    swap(arr[i], arr[j]);
            } //таким образом отсортируются четные элементы. далее рассмотрим
случай когда второй элемент нечетный
            else {
                if (!even(arr[i]) && arr[i] > arr[j])
                { /*здесь всё проще, так как менять элементы нужно только в
случае, когда
                                                    первый элемент тоже нечетный и он больше второго*/
                    swap(arr[i], arr[j]);
                }
            }
        }
    }

    print();
    system("pause");
    return 0;
}

```

Работа программы:

1) Ввод неправильного количества элементов массива

 Выбрать C:\Users\Marina Chebotova\source\repos\laba5\Debug\laba5.exe

Введите количество элементов:

hello

Значение не допустимо!

Введите количество элементов:

2) Работа программы

 Выбрать C:\Users\Marina Chebotova\Desktop\laba5\Debug\laba5.exe

Введите количество элементов:

9

Введите элементы:

Ввод 1 элемента

4

Ввод 2 элемента

7

Ввод 3 элемента

12

Ввод 4 элемента

7

Ввод 5 элемента

10

Ввод 6 элемента

3

Ввод 7 элемента

8

Ввод 8 элемента

23

Ввод 9 элемента

1

Массив успешно считан.

Вывод массива...

4 8 10 12 1 3 7 7 23 Для продолжения нажмите любую клавишу . . .