

Задание 9-10

Сортировки вставками

1) Выполнение данной работы продолжается в программе 2016_SAS05_Фамилия, где фамилия – фамилия автора программы на латинице.

2) Программа должна использовать функции, разработанные непосредственно автором программы.

3) В рамках этой работы необходимо сделать следующее:

3.1) Добавить функцию заполнения массива для сортировки случайными числами (количество чисел задаётся пользователем).

Для генерирования случайных чисел используется генератор псевдослучайных чисел компьютера, который генерирует числа из заданного диапазона с равномерным распределением.

Функции для работы с генератором случайных чисел описаны в библиотеке `stdlib.h` и `time.h`.

Нас интересует две функции:

- функция **`void srand(int seed)`** инициализирует генератор случайных чисел и имеет один целочисленный параметр, который задаёт «начальную точку генерации» для генератора. Эту функцию нужно вызвать один раз в самом начале работы программы. Обычно эта функция вызывается так:

```
srand( (unsigned) time(NULL) );
```

Так запускается генератор случайных чисел, инициализированный текущим значением времени. Таким образом, при каждом запуске программы, будут генерироваться разные значения.

- функция **`int rand(void)`** генерирует и возвращает случайное число от 0 до значения предопределенной константы `RAND_MAX`, которая

имеет значение 32767. Для генерации чисел из пользовательского диапазона от a до $b \leq 32767$ очевиден такой вызов:

```
rand() % (b - a) + a;
```

3.2) Добавить в алгоритмы сортировки вычисление времени сортировки, подсчет количества сравнений и перемещений.

Для вычисления времени, затрачиваемой на сортировку, будем исходить из следующего: если записать время перед выполнением сортировки и затем записать время после выполнения сортировки, то разницей будет время, затраченное на сортировку. Для записи значения времени будем использовать следующую функцию:

- функция `clock_t clock(void)` описана в библиотеке `time.h` и возвращает, сколько процессорного времени было затрачено на программу, из которой вызывается функция `clock()`. То есть учитывается не общее время работы программы, а то время, которое процессор потратил на обработку конкретно этой программы.

Таким образом, нам понадобятся две переменные `start` и `finish` типа `clock_t`, в которые будем записывать значение, возвращаемые функцией `clock()` перед началом сортировки, и после сортировки. Так как функция возвращает количество процессорного времени (количество тактов работы процессора – *ticks of the internal clock*), для получения времени в секундах, нужно выполнить перевод из тактов в секунды. Константа `CLOCKS_PER_SEC`, определенная в библиотеке `time.h`, содержит количество тактов в секунду. Для получения значения в секундах нужно выполнить следующее: `duration = (double)(finish - start) / CLOCKS_PER_SEC;`

Для подсчета количества сравнений и перестановок введём новые переменные и будем увеличивать их значение при каждом сравнении или

перестановке. При оценке общего времени сортировки, временем, затрачиваемым на увеличение значения этих переменных на 1 можно пренебречь.

3.3) Реализовать сортировку вставками.

3.4) Внедрить вышеперечисленные операции в уже реализованные алгоритмы сортировки (в сортировки подсчётом – только учёт времени)

3.5) Реализовать сортировку Шелла.

Оценка выполнения задания производится следующим образом:

Оценка 5 (отлично): все пункты задания выполнены полностью, программа является работоспособной, исполнитель обоснованно отвечает на контрольные вопросы.

Оценка 4 (хорошо): все пункты задания выполнены полностью, программа является работоспособной, исполнитель хорошо отвечает на контрольные вопросы, но не всегда может обосновать их.

Оценка 3 (удовлетворительно): пункты задания выполнены частично, программа является частично работоспособной, исполнитель затрудняется в ответах на контрольные вопросы.

Оценка 2 (неудовлетворительно): пункты задания выполнены частично, программа не является работоспособной, исполнитель затрудняется в ответах на контрольные вопросы.

Соответствие оценок баллам балльно-рейтинговой системы при оценивании выполнения практической работы:

Оценка 5 (отлично) – 2 балла;

Оценка 4 (хорошо) – 1,5 балла;

Оценка 3 (удовлетворительно) – 1 балл;

Оценка 2 (неудовлетворительно) – 0 баллов.