Методы разработки функциональных программ.

**Задание 1**.

Написать программу сортировки списка методом Шелла. Вычисление последовательности шагов сортировки производится в соответствии с вариантом в Таблице 1:

Методом Р. Седжвика, рассмотренным в [6].

Результат:

"Lab 3: task 1"

(1 3 3 4 5 6 8 8 9 9 10 11 12 13 14 18)

[Finished in 0.087s]

Обоснование:

Функция sort\_() выполняет сортировку методом Шелла по возрастанию. Для того, чтобы выполнить сортировку по убыванию, достаточно в функции pass\_lst\_() в условии системного вызова if() сменить знак “<“ на “>”.

Функция even\_() проверяет число на четность с помощью вычисления остатка от деления благодаря системному вызову mod().

**Задание 2.**

Написать программу сортировки [6] списка в соответствии с вариантом в таблице 2:

Сортировка методом прямого выбора.

Сравнить эффективность реализованной сортировки и реализованного в Задании 1 варианта сортировки Шелла.

Результат:

"Lab 3: task 2"

(18 14 13 12 11 10 9 9 8 8 6 5 4 3 3 1)

[Finished in 0.045s]

Обоснование:

Для сортировки методом прямого выбора, помимо написания основной функции сортировки sort\_(), понадобилось написать следующие функции:

max\_() — для поиска максимального элемента в списке;

swap\_() — для замены элементов списков местами.

Реализация данной сортировки ощутимо проще, чем сортировка методом Шелла и скорость её выполнения гораздо быстрее. В этом можно убедиться, сравнив время выполнения двух программ.

**Задание 3.**

Написать программу объединения двух отсортированных списков в один. При этом порядок сортировки в списке-результате должен сохраняться.

Результат:

"Lab 3: task 3"

(18 14 13 12 11 10 9 9 8 8 6 5 4 3 3 1)

[Finished in 0.053s]

Обоснование:

Для выполнения этого задания потребовалось позаимствовать функции max\_() и swap\_() и sort\_() из предыдущего. Объединение двух списков в один производилось с помощью их рекурсивных вызовов.

**Задание 4.**

Написать программу в соответствии с заданием из Таблицы 3:

Написать программу, возвращающую список, содержащий информацию о количестве подсписков на каждом уровне вложенности : ((<уровень><количество подсписков>)...).

Результат:

"Lab 3: task 4"

((3 1) (2 2) (1 3))

[Finished in 0.056s]

Обоснование:

Запись (3 1) означает, что в этом списке функция нашла 3 подсписка:

(8 2), (1 (2 (5 0)) (3 1)), (2 5);

Запись (2 2) означает, что во 2-ом подсписке находятся еще 2 подсписка:

(2 (5 0)), (3 1);

Запись (1 3) означает, что во 3-ем подсписке находится 1 подсписок:

(2 5)