Отчет по лабораторной работе (Вариант 4)

• Цель работы

Данная работа предполагает изучение и практическую реализацию задачи по перечислению последовательностей генов, которая в контексте биоинформатики связана с анализом перестроек генома и сравнением синтенных блоков у разных видов. Такая задача математически эквивалентна перечислению всех перестановок заданной длины. Таким образом, основная цель – разработать программу, способную находить общее число перестановок натуральных чисел от 1 до *n* и выводить список всех таких перестановок.

• Задачи

1. Прочитать входное данное – натуральное число *n*, представляющее длину перестановок.
2. Вычислить общее число перестановок длины *n*. Для набора из *n* различных элементов (чисел от 1 до *n*) количество перестановок равно *n*!.
3. Сгенерировать все возможные упорядоченные последовательности натуральных чисел от 1 до *n*, где каждое число используется ровно один раз. Эти последовательности и есть перестановки длины *n*.
4. Вывести полученное общее число перестановок.
5. Вывести список всех сгенерированных перестановок.

• Инструменты, Алгоритмы

Для реализации задачи по генерации перестановок был использован алгоритм, предоставляемый стандартной библиотекой Python. В частности, был задействован модуль itertools и его функция permutations().

Алгоритм реализован следующим образом:

1. Считывается входное натуральное число n. (input())
2. Используется функция itertools.permutations(range(1, n+1)) для создания итератора, который генерирует все перестановки последовательности чисел от 1 до n (включительно). range(1, n+1) создает последовательность чисел 1, 2, ..., n.
3. Результат работы itertools.permutations преобразуется в список (list2) для удобства подсчета и последующего вывода.
4. Общее число перестановок определяется как длина полученного списка (len(list2)) и выводится на экран.
5. Каждая перестановка из списка list2 выводится построчно, при этом элементы перестановки объединяются в строку с пробелами между ними с помощью map(str, finish) и ' '.join(...).

• Ошибки и их исправления

---

(Данная работа была написана мною довольно давно. По моей памяти, каких-либо ошибок или затруднений не возникало)

• Выводы

Выполнение данного варианта лабораторной работы позволило успешно реализовать алгоритм генерации и перечисления всех перестановок для заданного натурального числа *n*. Используя стандартные возможности языка Python, в частности модуль itertools, была получена программа, которая корректно вычисляет общее количество перестановок и выводит их список.