ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информатика и вычислительная техника»

Отчёт по лабораторной работе № 1

по дисциплине

«Проектирование и тестирование программного обеспечения»

Выполнил:

Студент гр. ПИН-222

Лубенский И. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подп., дата)

Проверил:

Старший преподаватель каф. ИВТ

Блохин А. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подп., дата)

Омск, 2024

**ОПИСАНИЕ ЗАДАЧИ “3N + 1”**

Полагают (но это еще не доказано), что алгоритм 3n + 1 сведется к n = 1 для любого целого n. По крайней мере, это предположение верно для всех целых чисел до 1 000 000.

Задача 3n + 1 работает с введённым целым числом n. Если число чётно, то оно делится на 2. Если нечётно, то умножается на 3 и к полученному значению прибавляется 1. Этот процесс повторяется с каждым новым значением до тех пор, пока из него не будет получено значение 1.

В ходе преобразований получается некоторая последовательность чисел. Например, для n = 22 будет сгенерирована следующая последовательность:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Для введённого n длиной цикла называется число сгенерированных чисел до и включая 1. То есть в примере с n = 22 длина цикла равна 16.

Для двух заданных i и j необходимо определить максимальную длину цикла для всех чисел между i и j, включая обе конечные точки.

Входные данные будут состоять из серии пар целых чисел i и j, одна пара чисел в строке. Все целые числа будут меньше 1 000 000 и больше 0.

Для каждой пары чисел i и j выведите i, j в том порядке, в каком они были введены, и после этого выведите максимальную длину цикла для всех целых чисел между i и j включая сами i и j. Эти три числа должны быть разделены одним пробелом, все три числа в одной строке, и для каждой строки входных данных должна быть одна строка выходных данных.

Пример входных данных Соответствующие выходные данные

1 10 1 10 20

100 200 100 200 125

201 210 201 210 89

900 1000 900 1000 174

**АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ “3N + 1”**

Схема алгоритма решения поставленной задачи показана на рисунке 1.

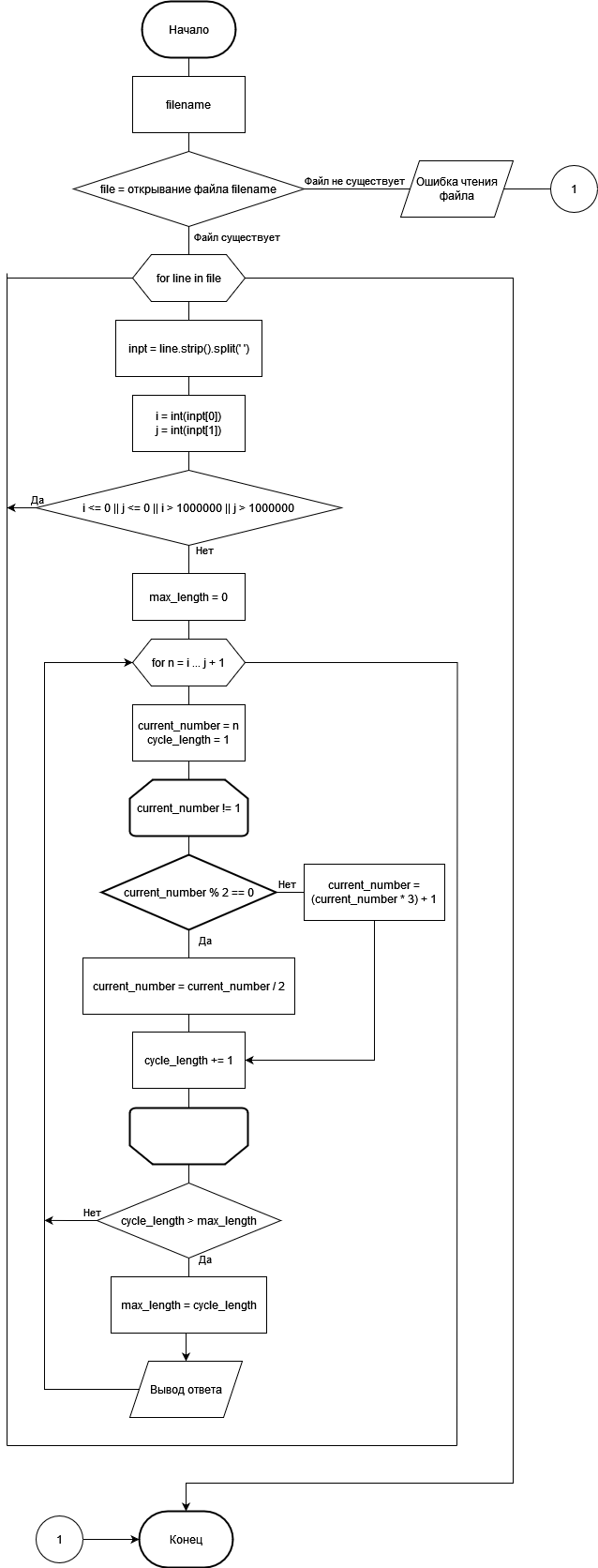


Рисунок 1 — Схема составленного алгоритма

Исходный код программы показан на рисунке 2.

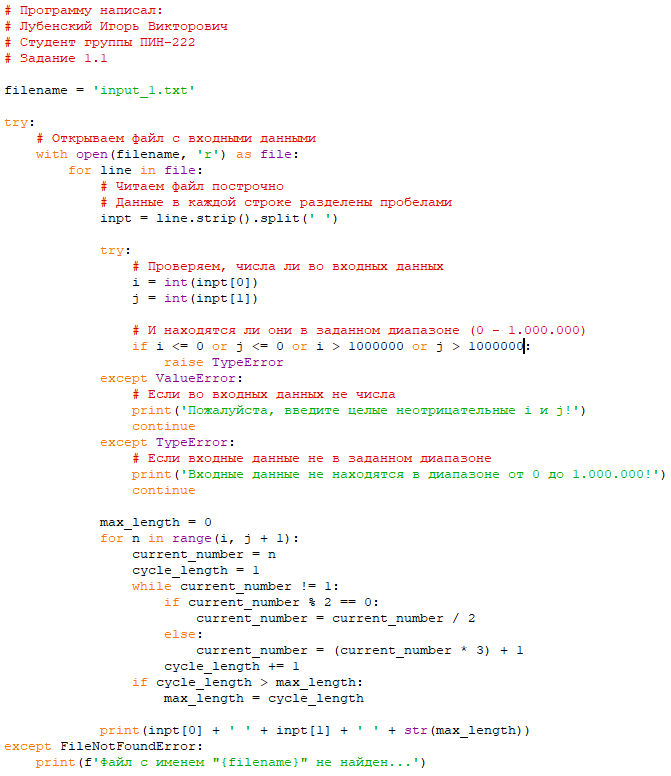


Рисунок 2 — Исходный код для решения задачи

В качестве языка программирования для решения задачи был выбран язык программирования Python.

Программа пытается открыть файл с указанным названием. Если это не удаётся сделать, то выводится ошибка о том, что файл не найден. Если файл удалось открыть, то входные данные из него считываются построчно. Из каждой строки убираются лишние пробелы, данные в каждой строке разделяются пробелами.

После считывания очередной строки программа пытается привести входные данные к типу данных int, так как по условию задачи значения i и j должны быть целыми. Кроме того, входные данные также проверяются на соответствие диапазону от 0 до 1 000 000, указанному в условии задачи. Если данные из файла не удалось привести к целым числам или они не входят в заданный диапазон, то программа выведет соответствующие ошибки.

Если входные данные в конкретной строке были валидными, то начинается работа алгоритма 3n + 1. На каждой итерации происходит подсчёт длины последовательности, которая по умолчанию равна 1.

Для каждой строки выводится число i, число j, а также длина последовательности, получаемой при отработке алгоритма от i до j. Если входные данные в конкретной строке были невалидными, то для данной строки будет выведена соответствующая ошибка и программа приступит к обработке следующей строки файла.

**ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

На рисунке 3 показаны входные данные, находящиеся в файле.

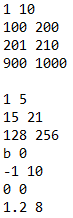


Рисунок 3 — Входные данные

Первые четыре строки входных данных были скопированы из методических указаний. Далее следуют тесты, обрабатывающие крайние случаи, намеренные ошибки во входных данных (пустые строки и символы), а также дробные числа, недопустимые по условию задачи.

Результат обработки указанных входных данных показан на рисунке 4.

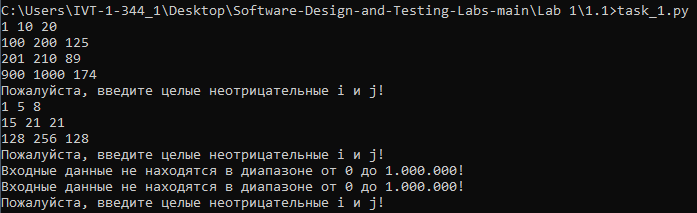


Рисунок 4 — Результат работы программы

Результаты обработки первых четырёх строк сходятся с методическими указаниями, то есть непосредственно алгоритм был реализован верно. Пустая строка была обработана программой верно — она попросила ввести числовые данные. Следующие три теста были обычными валидными данными, которые программа успешно обработала по алгоритму. Последние четыре теста завершились с ошибками: в первом случае на вход поступил символ “b”, который было невозможно обработать алгоритмом, далее поступило две строки данных, не входящих в заданный диапазон от 0 до 1 000 000, а в последней строке находилось число 1.2, которое не является целым, что недопустимо по условию задачи.

**ОПИСАНИЕ ЗАДАЧИ “JOLLY JUMPERS”**

Последовательность n > 0 целых чисел называется Jolly Jumper, если абсолютные значения разностей последовательных элементов принимают все возможные значения от 1 до n - 1. К примеру,

1 4 2 3

это Jolly Jumper, потому что абсолютные разности равны 3, 2 и 1 соответственно. Определение подразумевает, что любая последовательность из одного числа — это Jolly Jumper. Напишите программу, которая определяет, является ли каждая из введенных последовательностей Jolly Jumper.

Каждая строка входных данных содержит число n < 3000, за которым следуют n целых чисел, представляющих собой последовательность.

Для каждой строки входных данных выведите строку, говорящую "Jolly" или "Not Jolly".

Пример входных данных

4 1 4 2 3

5 1 4 2 -1 6

**АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ “JOLLY JUMPERS”**

Схема алгоритма решения поставленной задачи показана на рисунке 5.

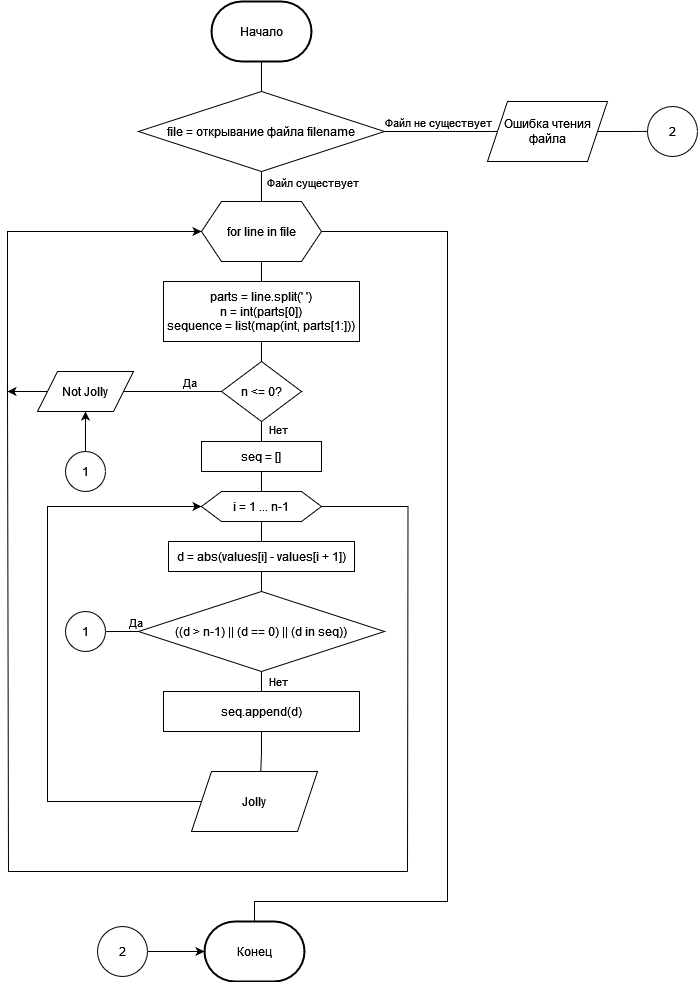


Рисунок 5 — Схема алгоритма решения задачи Jolly Jumper

Исходный код программы показан на рисунке 6.

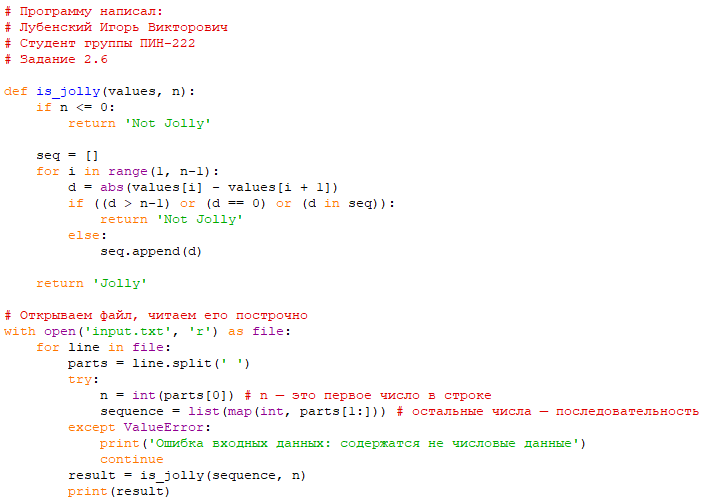


Рисунок 6 — Исходный код решения задачи

В программе создана специальная функция для проверки, является ли последовательность из n значений Jolly Jumper.

Если во входных данных в качестве n было отрицательное число, то функция сразу сообщает, что последовательность Not Jolly, потому что в данных сказано, что задана последовательность из отрицательного количества чисел, что невозможно.

В функции создаётся массив для хранения абсолютных значений разностей последовательных значений в наборе. Если абсолютная разность каких-либо значений выходит больше, чем n -1 или получается равной 0, или получается число, которое уже было среди абсолютных разностей, то программа говорит, что данная последовательность Not Jolly. Иначе полученное значение абсолютной разности добавляется в последовательность, и обработка продолжается. Если условия соблюдены, то функция вернёт строку “Jolly” для заданного n и последовательности чисел. Эта функция используется внутри основной части кода.

Программа открывает файл с входными данными и считывает его построчно. Данные в строках разделяются по пробелам. Первым элементом в строке является n, оно приводится к числу. Остальные элементы являются последовательностью, они также приводятся к числу. Если какие-либо данные не удалось привести к числовым, то программа выводит соответствующую ошибку. Если данные были успешно приведены к необходимому типу данных, то они передаются в функцию, описанную выше, и обрабатываются ей.

**ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

На рисунке 7 показаны входные данные, находящиеся в файле.

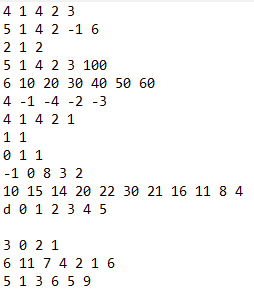


Рисунок 7 — Входные данные для задачи Jolly Jumper

Первые две строчки входных данных были скопированы из методических указаний. Остальные данные являются либо валидными, либо специально неправильными, либо содержат текстовые данные или пустые данные.

Результат обработки входных данных показаны на рисунке 8.

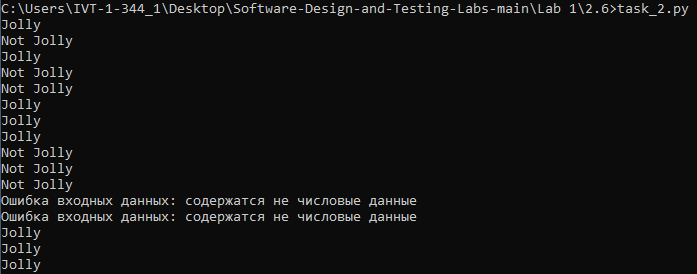


Рисунок 8 — Результат работы программы

В программе верно построен основной алгоритм определения Jolly и Not Jolly последовательностей. Это видно из того, что выходные данные программы сходятся с примером из методических указаний.

В строке 12 входных данных в качестве значения n специально был указан символ “b”, который невозможно интерпретировать как целое число, о чём программа успешно сообщила и перешла к обработке следующих входных данных.

Строка 13 была оставлена пустой, что также было верно обработано программой и была выведена соответствующая ошибка.