Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №1**

Выполнили:

Мануковская Д. М.

Сакулин И. М.

Проверил:

Мусаев А. А.

Санкт-Петербург,

2024

**Содержание**

[Введение 3](#_Toc177491545)

[Задание 1 4](#_Toc177491546)

[Задание 2 6](#_Toc177491547)

[Задание 3 10](#_Toc177491548)

[Заключение 11](#_Toc177491549)

Введение

В данной лабораторной работе необходимо реализовать программу, выводящую количество шагов, которые понадобятся бинарному поиску для нахождения нужного числа. А также нужно реализовать код для поиска студента факультета по его характеристикам. На основе этой программы будет создан граф.

Цель работы: знакомство с графами.

Задание 1

Необходимо реализовать бинарный поиск. Результатом программы является количество шагов, которые будут выполнены для нахождения загаданного числа.

Функция bin\_s (рисунок 1) бинарным поиском находит нужный элемент в переданном диапазоне и возвращает число шагов, необходимых для этого.

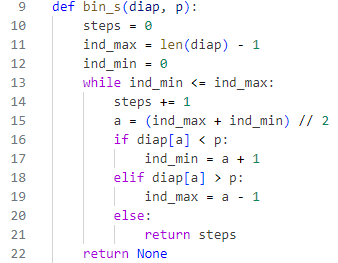


Рисунок 1 – Функция bin\_s

В функцию передаются список/диапазон, в котором находится загаданный элемент, и само это число. Счетчик steps увеличивается на 1 с каждым прохождением цикла. Переменная a – это индекс предполагаемого загаданного элемента. В зависимости от значения предполагаемого и загаданного значений изменяется диапазон поиска, т. е. ind\_max или ind\_min.

У пользователя запрашивается начало, конец и шаг нужного списка. Если введённые значения неверны, выводиться комментарий. Это осуществляется за счет конструкции try-except и ранее созданных классов ошибок (рисунок 2).

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 2 – Классы ошибок

Основной код представлен на рисунке 3.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 3 – Основной код

Взаимодействие с пользователем осуществляется через консоль. На рисунке 4 выведены все возможные ситуации работы программы.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 4 – Работа программы

Задание 2

Для своей учебной группы составить словарь, который будет описывать характеристики каждого из студентов. Реализовать программу, которая по определенным характеристикам будет угадывать студента.

Для составления характеристик студентов был проведен опрос, в котором поучаствовали 34 человека, в том числе адаптеры. Для удобства работы полученная информация была переформатирована в словарь (рисунок 5): ключи – характеристики, представленные в виде двоичного кода, значения – ФИО каждого опрошенного.

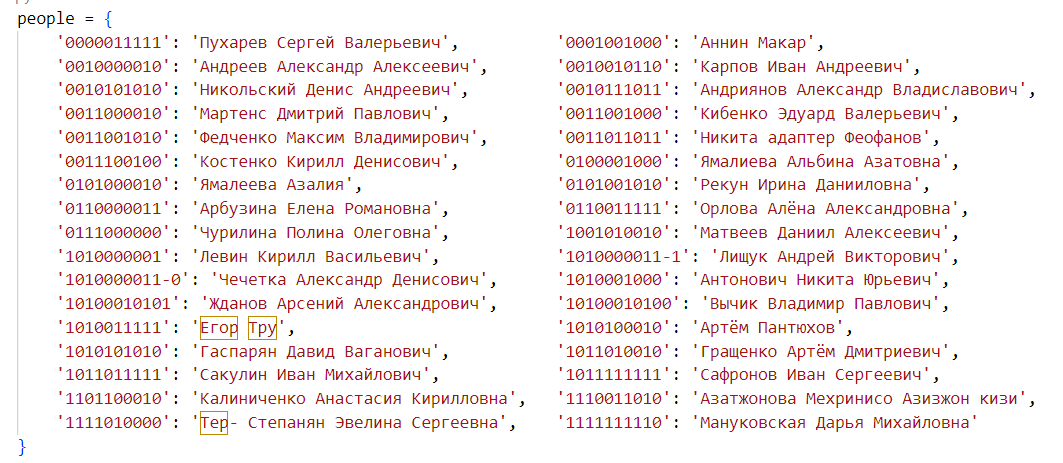


Рисунок 5 – Словарь характеристик

Каждое значение «1» или «0» обозначает ответ «да» или «нет» соответственно, значения установлены в том же порядке, что и вопросы. Список всех вопросов был записан в список, представленный на рисунке 6.

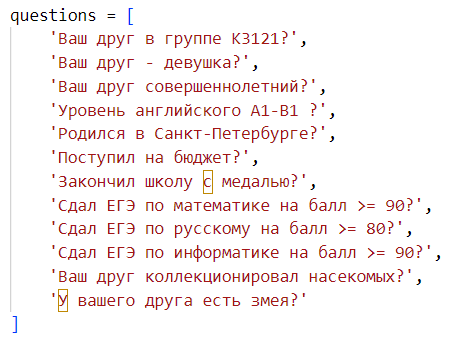


Рисунок 6 – Список вопросов

Таким образом, ответ на вопрос, который находиться по индексу 0 в списке questions храниться по тому же индексу 0 в словаре people.

Затем были составлены множества ответов с положительным и отрицательным значением, то есть аналоги слов да и нет (рисунок 7).

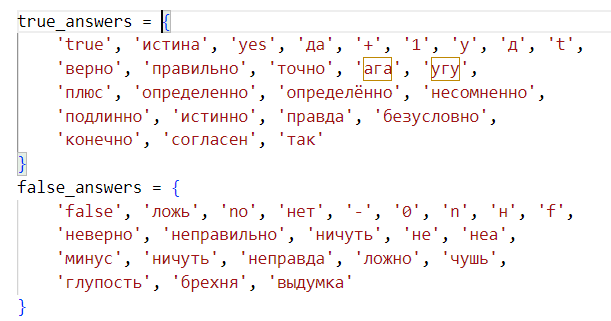


Рисунок 7 – Вариации ответов

Все списки и словарь, представленные выше, были записаны в файл «data.py». В основном коде они были импортированы через «from data import \*».

Первым делом был написан код, который запрашивает у пользователя значения и приводит их к типу Boolean (рисунок 8).

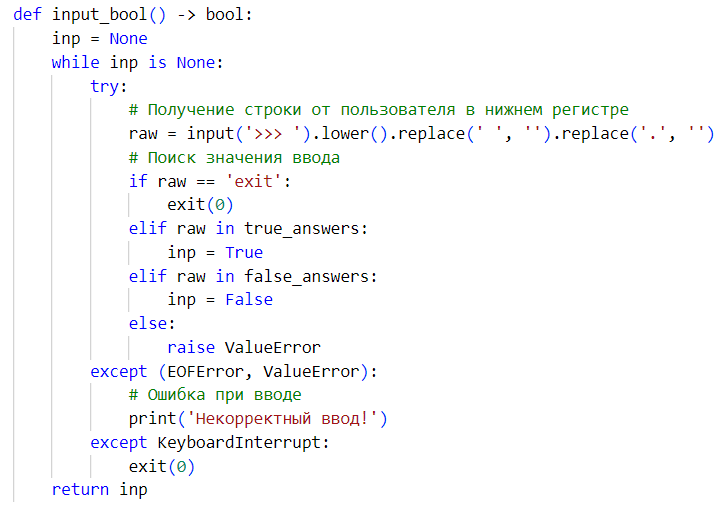


Рисунок 8 – Получение ответа пользователя

Перейдём к основной части программы (рисунок 9). Сначала на экран выводиться приветствие и инструкция для пользователя. Затем все действия будут происходить в бесконечном цикле, в котором будет изменяться только переменная состояния «step». Первым делом, чтобы избежать потенциальные ошибки, проверяется не вышла ли переменная состояния за пределы списка вопросов, а если это всё-таки произойдёт, игра перезапустится. Это также произойдёт при первом входе в цикл. Внутри блока «Запуск» происходит переход к состоянию 0, заполнение списка ключей-характеристик и вывод сообщения о начале игры.

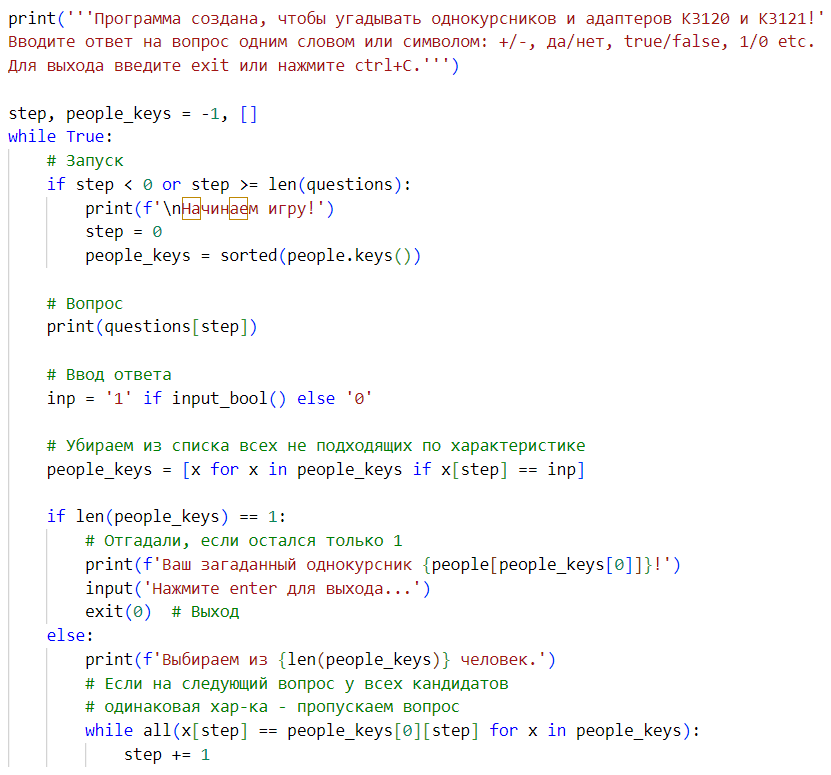


Рисунок 9 – Основной алгоритм игры

Далее задаётся вопрос, который находиться по индексу «step» и получается ответ от пользователя. Оставляются в списке ключей-характеристик только те, которые подходят под ответ (т.е. подходят все значения с 0 по «step»).

Если на каком-то шаге в списке остаётся всего 1 человек, значит он и является искомым – отображается на экран его имя и программа завершается. В противном случае происходит переход на следующий шаг, минуя те шаги, которые не дадут новой информации (т.е. не сузят список ключей-характеристик)

Результат работы программы представлен на рисунке 10.

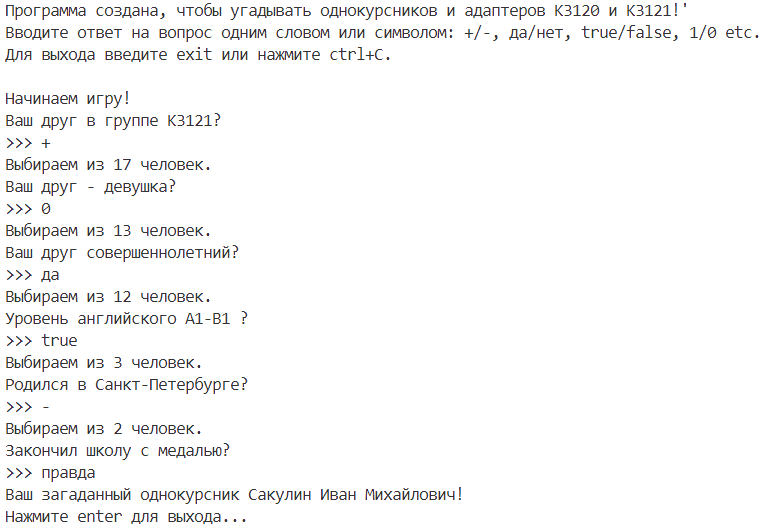


Рисунок 10 – Работа программы

Задание 3

Составьте граф для задания №2. К какому типу относится данный граф?

Был составлен граф по заданию 2 (рисунок 11). Этот граф представляет собой дерево с разветвлением исходя из ответов да или нет на поставленный вопрос. Он наглядно показывает, что каждый студент имеет характеристику отличную от других, т. е. характеристики, представленные в виде двоичного кода (ответ да – 1, нет – 0), являются уникальными и есть возможность использовать их в коде как id или ключи словаря.

Изображение выглядит как текст, снимок экрана, диаграмма, дизайн

Автоматически созданное описание

Рисунок 11 – Граф

Заключение

Цель работы достигнута. Был реализован бинарный поиск и программа, вычисляющая количество шагов для нахождения загаданного элемента массива. В ходе работы создан код для поиска студентов, а также составлен словарь, описывающий характеристики обучающихся, и сделан граф в виде дерева для него.