1. Напишите игру “Викторина”. Вопросы и ответы должны храниться в текстовом файле.  
   Перед непосредственным написанием кода определим структуру файла, в котором будут храниться вопросы для игры.  
   Структура файл:

<Вопрос №1>

<ответ 1>

<ответ 2>

<ответ 3>

<ответ 4>

<правильный ответ>

<Вопрос №2>

….

Пример файла с вопросами:

В какую из этих игр играют не клюшкой  
Хоккей  
Гольф  
Поло  
Бильярд  
Бильярд  
С какой из этих стран Чехия не граничит?  
Германия  
Чехия  
Австрия  
Польша  
Чехия  
Как называется комедия В. В. Маяковского?  
Пена  
Жук  
Клоп  
Паук  
Территория какой из этих стран - наибольшая?  
Япония  
Германия  
Италия  
Финляндия  
Япония

После определения структуры файла перейдем в IDE и начнем разработку приложения.

import random  
  
  
# Функция, которая создает словарь для хранения вопроса, списка вариантов и правильного ответа  
# question - это вопрос одного раунда игры  
def create\_sample(question):  
 return {  
 "question": question,  
 "variants": list(),  
 "answer": ""  
 }  
  
  
# Функция для считывания данных из текстового файла  
# Возвращает список словарей, каждый словарь создан с помощью функции create\_sample()  
def read\_from\_file(path\_file):  
 # Список для хранения словарей  
 data = list()  
 # Пример из файла, который считывается из файла, изначально None  
 sample = None  
 with open(path\_file, 'r', encoding='utf8') as f:  
 # Для удобного подсчета количества считанных строк используем функцию enumerate. enumerate принимает   
 # структуру данных, которую можно использовать в цикле for .. in.  
 # enumerate возвращает кортеж:  
 # (index, obj), где index - порядковый номер obj начиная с 0  
 # obj - элемент структуры данных  
 for i, line in enumerate(f):  
 # Удаляем из считанной строки символ перевода строки "\n"  
 line = line.replace("\n", "")  
 # Если остаток от деления на 6 равен 0, тогда создаём новый словарь   
 if i % 6 == 0:  
 sample = create\_sample(line)  
 # Считываем 4 варианта ответа  
 elif i % 6 < 5:  
 sample["variants"].append(line)  
 else:  
 # Считываем правильный вариант ответа   
 sample['answer'] = line  
 # Добавлем полученный словарь в список данных для игры  
 data.append(sample)  
 return data  
  
  
# Функция для получения данных следующего раунда игры  
# Удаляет выбранный вариант  
def next\_question(data):  
 # random.choice() - возвращает случайный элемент из последовательности  
 sample = random.choice(data)  
 data.remove(sample)  
 return sample  
  
  
# Функция для проверки ответа пользователя  
# user\_id\_answer - номер варианта, который выбрал пользователь  
# sample - текущий раунд (словарь)  
def check\_answer(user\_id\_answer, sample):  
 # Не забываем сделать -1 из ответа пользователя, т.к нумерация в листе начинается с 0  
 user\_answer = sample["variants"][user\_id\_answer - 1]  
 print(sample["variants"])  
 return user\_answer == sample["answer"]  
  
  
# Функция для вывода вариантов ответов на текущий вопрос  
def print\_variants(sample):  
 print("Варианты ответа")  
 for i, var in enumerate(sample['variants']):  
 # В столбец выводим варианты ответов, нумерация начиная с 1  
 print(i + 1, ". ", var)  
  
  
# Функция для проверки нужно ли продолжить игру  
def check\_continue():  
 answer = input("Следующий вопрос?[Д/Н]\n")  
 return answer.strip().lower() == 'д'  
  
  
# Функция для организации игрового процесса  
# Считываем данные из файла  
# Функция работает пока пользователь не ответит на все вопросы или не откажется играть дальше  
def main():  
 tasks = read\_from\_file('quiz.txt')  
 print("Добро пожаловать в игру Викторина!!!!")  
 round\_number = 1  
 while True:  
 sample = next\_question(tasks)  
 print("Вопрос # ", round\_number)  
 print(sample["question"])  
 print\_variants(sample)  
 id\_answer\_user = int(input("Введите номер варианта: "))  
 is\_right = check\_answer(id\_answer\_user, sample)  
 if is\_right:  
 report\_result = "Вы ответили правильно!"  
 else:  
 report\_result = "Вы ответили не правильно! Правильный ответ: " + sample['answer']  
 print(report\_result)  
 if len(tasks) == 0:  
 print("Викторина закончена, пока!")  
 break  
 is\_continue = check\_continue()  
 if not is\_continue:  
 break  
 round\_number += 1  
  
  
main()

1. Напишите программу для создания файлов раундов игры “Викторина”. Один раунд это:

вопрос, варианты ответов, правильный ответ. Выходным файлом программы должен быть бинарный файл (файл с расширением .bin)

import pickle  
  
  
# Функция, которая создает словарь для хранения вопроса, списка вариантов и правильного ответа  
# question - это вопрос одного раунда игры  
def create\_sample():  
 return {  
 "question": "",  
 "variants": list(),  
 "answer": ""  
 }  
  
  
# Функция для создания одного раунда игры Викторина  
def create\_round():  
 sample = create\_sample()  
 sample["question"] = input("Введите вопрос: ")  
 for i in range(0, 4):  
 print(i + 1, ".Введите вариант ответа: ")  
 variant = input().strip()  
 sample["variants"].append(variant)  
 index\_correct\_answer = int(input("Введите индекс правильного ответа:")) - 1  
 sample['answer'] = sample["variants"][index\_correct\_answer]  
 return sample  
  
  
# Функция для записи всех раундов в бинарный файл с помощью модуля pickle  
def write\_to\_bin(data, file\_name):  
 with open(file\_name + ".bin", "wb") as f:  
 pickle.dump(data, f)  
  
  
# Функция для проверки, хочет ли пользователь создать еще раунд для игры  
def check\_continue():  
 answer = input("Создать еще раунд?[Д/Н]")  
 return answer.lower().strip() == "д"  
  
  
# Функция для организации работы программы  
# Создаем список словарей. Заполняем этот список раундами, которое придумал пользователь  
# Результат записываем в бинарный файл с заданным именем  
def main():  
 print("Вас приветствует программа для создания раундов игры Викторина")  
 file\_name = input("Введите имя файла, в который будет сохранен результат работы программы: ")  
 data = []  
 while True:  
 sample = create\_round()  
 data.append(sample)  
 if not check\_continue():  
 break  
 write\_to\_bin(data, file\_name)  
  
  
main()

1. Доработайте игру “Викторина” таким образом, чтобы в конце игры пользователю выводилось количество верно отвеченных вопросов и общее число вопросов.
2. Доработайте игру “Викторина” таким образом, чтобы в качестве файлов с данными использовался файл, полученный с помощью программы из задания 2.
3. Доработайте игру “Викторина” таким образом, чтобы она хранила в .csv файле список рекордов. В список должны попадать имя и результат игрока.
4. Реализуйте ту же самую функциональность, что и в предыдущей задаче, иным способом: на этот раз сохраните список рекордов в файле .xlsx.