Чтение файлов .seg

Прочитаем текст из файла .seg, выведем его построчно на экран. Какая информация содержится в каждой строке? Какой формат записи?

```
!wget https://pkholyavin.github.io/mastersprogramming/cta0001.seg_B2
```

```
with open("cta0001.seg_B2", "r", encoding="utf-8-sig") as f:
    lines = f.readlines()

for i, line in enumerate(lines):
    print(f"{i}\t{line.strip()}")
```

Соответствие уровней:

```
from itertools import product
letters = "GBRY"
nums = "1234"
levels = [ch + num for num, ch in product(nums, letters)]
print(levels)
level_codes = [2 ** i for i in range(len(levels))]
print(level_codes)
```

```
level2code = {i: j for i, j in zip(levels, level_codes)}
code2level = {j: i for i, j in zip(levels, level_codes)}
```

Хак для определения кодировки:

```
def detect_encoding(file_path):
    encoding = "utf-8"
    try:
        text = open(file_path, 'r', encoding="utf-8").read()
        if text.startswith("\ufeff"):  # т.н. byte order mark
            encoding = "utf-8-sig"
    except UnicodeDecodeError:
        try:
            open(file_path, 'r', encoding="utf-16").read()
            encoding = "utf-16"
        except UnicodeError:
            encoding = "cp1251"
    return encoding
```

(в общем случае это не работает, например:

```
with open("lol.txt", "w", encoding="cp1251") as f:
    f.write("Пё")


with open("lol.txt", "r", encoding="utf-8") as f:
    print(f.read())  # Greek Capital Letter Sho ???
```

)

Задания для выполнения в классе:

1. Напишите функцию, которая принимает на вход имя файла .seg и возвращает словарь, который содержит всю информацию из секции `[PARAMETERS]`. Не забудьте преобразовать данные в целочисленный тип!
   Пример словаря:

```
{
    "SAMPLING_FREQ": 22050,
    "BYTE_PER_SAMPLE": 2,
    "CODE": 0,
    "N_CHANNEL": 1,
    "N_LABEL": 13
}
```

2. Расширьте функцию так, чтобы она возвращала ещё и список меток. Сделайте каждую метку кортежем из трёх величин: позиция **в отсчётах**, уровень **в текстовом представлении**, имя метки (строка).
   Как перевести из номера байта в номер отсчёта? На что нужно разделить?
   Пример списка меток:

```
[
    (0, "B2", "j"),
    (4246, "B2", "u0"),
    (6354, "B2", "r'"),
    (6854, "B2", "i4 j"),
    (9090, "B2", "t"),
    (12452, "B2", "r'"),
    (12970, "B2", "i0"),
    (15772, "B2", "f"),
    (18403, "B2", "a4"),
    (19302, "B2", "n"),
    (20809, "B2", "a4"),
```

```
    (22254, "B2", "f"),
    (27331, "B2", "")
]
```

3. Модифицируйте функцию так, чтобы каждая метка была не кортежем, а словарём:

```
{"position": 0, "level": "B2", "name": "j"}
```

4. Выведите на экран все интервалы из файла .seg: попарно напечатайте первую и вторую метку, вторую и третью, третью и четвёртую, ..., предпоследнюю и последнюю.

5. Напишите функцию, которая принимает на вход словарь с параметрами и записывает первую половину файла .seg (секцию `[PARAMETERS]`)

6. Расширьте функцию так, чтобы она принимала на вход также и список меток и записывала файл .seg целиком

Домашнее задание:

1. Доработайте функции для чтения и записи файлов .seg: добавьте документацию, аннотации типов, постарайтесь учесть возможные ошибки (проверка файла на валидность)
2. Напишите программу, которая считывает файл .wav и параллельный ему файл .seg, делит файл .wav на интервалы, разграниченные метками из файла .seg, и записывает каждый фрагмент в отдельный файл, названный порядковым номером интервала и именем метки, открывающей соответствующий фрагмент.

Т.е. из файла `cta0001.wav` должны получиться, например:

```
0_j.wav
1_u0.wav
2_r'.wav  # одинарную кавычку можно заменить на что-нибудь другое, например, нижнее подчёр
...
```