```
import openpyxl
import time
from datetime import timedelta

"""
Badane zagadnienia
1. Liczba zwycięstw drugiego miejsca na liście nad ostatnim miejscem.
2. Liczba zwycięstw pierwszego miejsca na liście nad drugim miejscem.
3. Liczba zwycięstw pierwszego miejsca na liście nad ostatnim
miejscem.
4. Suma głosów analizowanych.
5. Suma głosów oddanych względem położenia na liście. Ustaliłem 7
następujących kategorii:
    a) Pierwsze miejsce;
    b) Drugie miejsce;
    c) Trzecie miejsce;
    d) Czwarte i piąte miejsca łącznie;
    e) Miejsca od szóstego do trzeciego od końca listy łącznie;
    f) Przedostatnie miejsce;
    g) Ostatnie miejsce.
6. Procent oddanych głosów na kandydatów względem położenia na liście.
Kategorie jak wyżej.
7. Stosunek głosów oddanych na pierwsze do głosów na ostatnie miejsce
na liście z podziałem na listy.
8. Stosunek głosów oddanych na drugie do głosów na ostatnie miejsce na
liście z podziałem na listy.
9. Stosunek głosów oddanych na pierwsze do głosów na drugie miejsce na
liście z podziałem na listy.
"""


def calculating_votes(sheet, cat):
    """
    Funkcja wykonuje różne obliczenia i testy na danych wyborczych
zapisanych w MS Excel Worksheet.
    :param sheet: sheet, Worksheet z danymi wyborczymi dla okręgu
    :param cat: list, lista o długości równej liczbie kategorii (7)
    :return: int, liczba głosów w okręgu, reszta informacji jest
zapisywana w globalnych strukturach
    """
    global WIN_LAST_OVER_FIRST
    # nazwiska kandydatów zaczynają się od czwartej komórki pliku xlsx
    col = 4
    # liczba głosów w danym okręgu
    votes_in_district = 0

    # Obliczenia są przeprowadzane dla każdej listy oddzielnie
    for cand_list in committees:
        # zmienna do przechowywania liczby głosów na każdego kandydata
danej listy
```

```
        votes_com = []
        # Wczytywanie i zapisywanie liczby głosów do czasu aż
natrafimy na kandydata innej listy
        # bądź nie ma już innych komitetów
        # Kandydaci są zapisani w pliku xlsx w formie KOWALSKI Jan -
KW ZZZZ
        while sheet.cell(row=1, column=col).value is not None and
cand_list in sheet.cell(row=1, column=col).value:
            votes_com.append(int(sheet.cell(row=2, column=col).value))
            col += 1  # przejście do następnej komórki

        # Testowy wydruk listy z głosami na kandydatów danego komitetu
        # print(f"{cand_list}: {votes_com}")

        # Porównywanie wyników z punktów 1-3
        if votes_com[1] > votes_com[-1]:
            comparison_2_last["Drugie"] += 1
        else:
            comparison_2_last["Ostatnie"] += 1
        if votes_com[0] > votes_com[1]:
            comparison_1_2["Pierwsze"] += 1
        else:
            comparison_1_2["Drugie"] += 1
        if votes_com[0] > votes_com[-1]:
            comparison_1_last["Pierwsze"] += 1
        else:
            # string z dodatkową informacją, która będzie wypisany
później
            WIN_LAST_OVER_FIRST = f"{sheet.title}: na liście
{cand_list} ostatni kandydat na " \
                                  f"liście uzyskał więcej głosów niż
pierwszy."
            comparison_1_last["Ostatnie"] += 1

        # sumowanie głosów potrzebne do obliczenia punktów 7-9
        votes_last_by_committee[cand_list] += votes_com[-1]
        votes_1_by_committee[cand_list] += votes_com[0]
        votes_2_by_committee[cand_list] += votes_com[1]

        # sumowanie wszystkich głosów w okręgu
        votes_in_district += sum(votes_com)

        # dodawanie odpowiednich wartości do listy przechowujących
liczbę głosów oddanych na poszczególne kategorie
        cat[0] += votes_com[0]
        cat[1] += votes_com[1]
        cat[2] += votes_com[2]
        cat[3] += sum(votes_com[3:5])
        cat[4] += sum(votes_com[5:len(votes_com)-2])
        cat[5] += votes_com[-2]
```

```
        cat[6] += votes_com[-1]

    # zwrócenie liczby głosów w okręgu, reszta informacji jest
zapisywana w globalnych strukturach
    return votes_in_district


def calculate_and_print_ratio(d1, d2):
    """
    Funkcja obliczająca stosunek głosów oddanych na jedno miejsce do
liczby głosów oddanych na inne miejsce
    z podziałem na komitety wyborcze.
    :param d1: Słownik typu {komitet: liczba_głosów_na_dane_miejsce}
    :param d2: Słownik typu {komitet: liczba_głosów_na_dane_miejsce},
jego wartości będą dzielić wartości d1
    """
    for key, val in d1.items():
        print(key + ' ' + str(round(val / d2[key], 2)))


# Zapisanie czasu rozpoczęcia programu
start_time = time.monotonic()

# Zmienna ze ścieżką do pliku .xlsx z wynikami wyborów
# Jest to workbook z 41 worksheetami, gdzie każdy worksheet zawiera
informacje o wynikach w jednym okręgu
FILE = "wyniki_gl_na_kandydatow_po_okregach_sejm_utf8.xlsx"

# W pliku listy.txt są zapisane komitety wyborcze, które zostały
zarejestrowane w każdym okręgu,
# łącznie 6 komitetów: BS, TD, NL, PiS, PO, Konfederacja
# Wczytywanie pliku, a dokładnie tylko pierwszej linii
with open("listy.txt", encoding="utf8") as f:
    line = f.readline()
    # metoda .readline() dodaje znak nowej linii na koniec stringa,
dlatego trzeba go usunąć ręcznie
    line = line[:-1]

# Wyodrębnienie komitetów ze wczytanych danych z pliku listy.txt i ich
wydruk
committees = line.split(",")
print("\nDostępne komitety:")
print(committees)

# ZMIENNE DO PRZECHOWYWANIA REZULTATÓW
# lista 7-elementowa, gdzie nr indeksu + 1 odpowiada miejscu kategorii
w punkcie 5
categories = [0] * 7

# zmienna do przechowania sumy głosów
```

```
all_votes = 0

# Słowniki przechowujące informacje o liczbie zwycięstw danego miejsca względem drugiego (punkty 1-3)
comparison_2_last = {"Drugie": 0, "Ostatnie": 0}
comparison_1_2 = {"Pierwsze": 0, "Drugie": 0}
comparison_1_last = {"Pierwsze": 0, "Ostatnie": 0}

# Słowniki przechowujące sumę głosów oddanych na poszczególne miejsca na każdej liście
votes_last_by_committee = {}
votes_1_by_committee = {}
votes_2_by_committee = {}
for item in committees:
    votes_last_by_committee[item] = 0
    votes_1_by_committee[item] = 0
    votes_2_by_committee[item] = 0

# Zmienna globalna do przechowywania informacji, w którym okręgu i na której liście ostatnie miejsce wygrało z pierwszym
WIN_LAST_OVER_FIRST = ""

# Otworzenie pliku xlsx z danymi za pomocą biblioteki openpyxl
data_workbook = openpyxl.load_workbook(filename=FILE)

# Zapisanie nazw wszystkich okręgów do jednej listy
electoral_districts = data_workbook.sheetnames

# Wykonanie obliczeń dla każdego okręgu
for district in electoral_districts:
    # print(district) - testowy wydruk okręgu
    # funkcja calculating_votes zwraca liczbę analizowanych głosów w danym okręgu, dodajemy ją do sumy wszystkich głosów
    all_votes += calculating_votes(data_workbook[district], categories)

# Wyniki punktu 1
print("\nLiczba zwycięstw drugiego miejsca na liście nad ostatnim miejscem:")
print(comparison_2_last)

# Wyniki punktu 2
print("\nLiczba zwycięstw pierwszego miejsca na liście nad drugim miejscem :")
print(comparison_1_2)

# Wyniki punktu 3
print("\nLiczba zwycięstw pierwszego miejsca na liście nad ostatnim miejscem :")
print(comparison_1_last)
```

```
print(WIN_LAST_OVER_FIRST)

# Wyniki punktu 4
print(f"\nLiczba analizowanych głosów: {all_votes}")

# Wyniki punktu 5
print("\nLiczba głosów oddanych na kandydatów względem położenia na
liście z podziałem na 7 kategorii:")
print(categories)

# Wyniki punktu 6
print("\nProcent głosów oddanych na kandydatów względem położenia na
liście z podziałem na 7 kategorii:")
results_categories = [round(i/all_votes*100, 2) for i in categories]
print(results_categories)

# Wyniki punktu 7
print("\nStosunek głosów oddanych na pierwsze miejsce na liście do
ostatniego z podziałem względem listy:")
calculate_and_print_ratio(votes_1_by_committee,
votes_last_by_committee)

# Wyniki punktu 8
print("\nStosunek głosów oddanych na drugie miejsce na liście do
ostatniego z podziałem względem listy:")
calculate_and_print_ratio(votes_2_by_committee,
votes_last_by_committee)

# Wyniki punktu 9
print("\nStosunek głosów oddanych na pierwsze miejsce na liście do
drugiego z podziałem względem listy:")
calculate_and_print_ratio(votes_1_by_committee, votes_2_by_committee)

# Obliczenie czasu trwania programu
end_time = time.monotonic()
print("\nCzas trwania programu:")
print(timedelta(seconds=end_time - start_time))


Dostępne komitety:
['KW BEZPARTYJNI SAMORZĄDOWCY', 'KKW TRZECIA DROGA PSL-PL2050 SZYMONA
HOŁOWNI', 'KW NOWA LEWICA', 'KW PRAWO I SPRAWIEDLIWOŚĆ', 'KW
KONFEDERACJA WOLNOŚĆ I NIEPODLEGŁOŚĆ', 'KKW KOALICJA OBYWATELSKA PO .N
IPL ZIELONI']

Liczba zwycięstw drugiego miejsca na liście nad ostatnim miejscem:
{'Drugie': 229, 'Ostatnie': 17}

Liczba zwycięstw pierwszego miejsca na liście nad drugim miejscem :
{'Pierwsze': 231, 'Drugie': 15}
```

```
Liczba zwycięstw pierwszego miejsca na liście nad ostatnim miejscem :
{'Pierwsze': 245, 'Ostatnie': 1}
Okręg wyborczy nr 32: na liście KW NOWA LEWICA ostatni kandydat na
liście uzyskał więcej głosów niż pierwszy.

Liczba analizowanych głosów: 21185794

Liczba głosów oddanych na kandydatów względem położenia na liście z
podziałem na 7 kategorii:
[8094767, 2837624, 2041342, 2433817, 4791250, 257744, 729250]

Procent głosów oddanych na kandydatów względem położenia na liście z
podziałem na 7 kategorii:
[38.21, 13.39, 9.64, 11.49, 22.62, 1.22, 3.44]

Stosunek głosów oddanych na pierwsze miejsce na liście do ostatniego z
podziałem względem listy:
KW BEZPARTYJNI SAMORZĄDOWCY 5.79
KKW TRZECIA DROGA PSL-PL2050 SZYMONA HOŁOWNI 9.47
KW NOWA LEWICA 7.63
KW PRAWO I SPRAWIEDLIWOŚĆ 9.94
KW KONFEDERACJA WOLNOŚĆ I NIEPODLEGŁOŚĆ 13.74
KKW KOALICJA OBYWATELSKA PO .N IPL ZIELONI 15.02

Stosunek głosów oddanych na drugie miejsce na liście do ostatniego z
podziałem względem listy:
KW BEZPARTYJNI SAMORZĄDOWCY 1.79
KKW TRZECIA DROGA PSL-PL2050 SZYMONA HOŁOWNI 3.49
KW NOWA LEWICA 3.45
KW PRAWO I SPRAWIEDLIWOŚĆ 4.06
KW KONFEDERACJA WOLNOŚĆ I NIEPODLEGŁOŚĆ 3.08
KKW KOALICJA OBYWATELSKA PO .N IPL ZIELONI 4.57

Stosunek głosów oddanych na pierwsze miejsce na liście do drugiego z
podziałem względem listy:
KW BEZPARTYJNI SAMORZĄDOWCY 3.24
KKW TRZECIA DROGA PSL-PL2050 SZYMONA HOŁOWNI 2.71
KW NOWA LEWICA 2.21
KW PRAWO I SPRAWIEDLIWOŚĆ 2.45
KW KONFEDERACJA WOLNOŚĆ I NIEPODLEGŁOŚĆ 4.46
KKW KOALICJA OBYWATELSKA PO .N IPL ZIELONI 3.29

Czas trwania programu:
0:00:00.297000
```