Concatenating Unlike-Structured Tables

Create a table that contains monthly public use statistics for 2014, 2015, and 2016 from the National Park Service.

1. Complete the Level 1 practice or submit the following code:

```
data work.np_combine;
    set pg2.np_2015 pg2.np_2016;
    CampTotal=sum(of Camping:);
    where Month in(6, 7, 8);
    format CampTotal comma15.;
    drop Camping:;
run;
```

2. Open the **pg2.np_2014** table and compare the column names with the **np_combine** table. Which column or columns in **np_2014** must be renamed to match columns in **np_combine**?

   The **Park** and **Type** columns must be renamed.

3. Modify the program.

   - Concatenate the **pg2.np_2014**, **pg2.np_2015**, and **pg2.np_2016** tables.
   - Rename the columns as necessary to align the columns with similar values.
   - In addition to filtering rows by **Month**, also include only rows where **ParkType** is *National Park*.
   - Arrange the newly created table in ascending order by **ParkType**, **ParkCode**, **Year**, and **Month**.
   - Submit the program and examine the output data.

```
data work.np_combine;
    set pg2.np_2014(rename=(Park=ParkCode Type=ParkType))
        pg2.np_2015
        pg2.np_2016;
    CampTotal=sum(of Camping:);
    where Month in(6, 7, 8) and ParkType="National Park";
    format CampTotal comma15.;
    drop Camping:;
run;


proc sort data=np_combine;
    by ParkType ParkCode Year Month;
run;
```

4. How many rows are in the **np_combine** table, and what is the value of **ParkCode** in row 1?

   The **np_combine** table has 531 rows, and the **ParkCode** value is *ACAD* in row 1.