```
1  ************************************************************;
2  *  LESSON 5, PRACTICE 5                                    *;
3  *  a) The first three steps sort and merge the             *;
4  *     PG2.NP_CODELOOKUP and PG2.NP_FINAL tables.           *;
5  *     Highlight the first two PROC SORT steps and the      *;
6  *     DATA step and run the selected code. Examine the     *;
7  *     highuse table.                                       *;
8  *  b) Add a subsetting IF statement in the DATA step to    *;
9  *     output only the rows in which DayVisits is greater   *;
10 *     than or equal to 5,000,000. Highlight the DATA step *;
11 *     and run the selected code. Why must you use IF       *;
12 *     instead of a WHERE statement?                        *;
13 *  c) Run the final PROC SORT step to sort and subset the *;
14 *     PG2.NP_SPECIES table. Compare the columns in the     *;
15 *     output birds table with the highuse table to         *;
16 *     determine the matching column.                       *;
17 *  d) Add a PROC SORT step to sort the highuse table by    *;
18 *     the matching column in the birds table.              *;
19 *  e) Add a DATA step to merge the highuse and birds       *;
20 *     tables and create a table named BIRDS_LARGEPARK.     *;
21 *     Include in the output table only ParkCode values     *;
22 *     that are in the highuse table.                       *;
23 ************************************************************;
24
25
26 /*Steps a and b*/
27 proc sort data=pg2.np_CodeLookup
28           out=work.sortedCodes;
29     by ParkCode;
30 run;
31
32 proc sort data=pg2.np_2016
33           out=work.sorted_code_2016;
34     by ParkCode;
35 run;
36
37 data work.parkStats(keep=ParkCode ParkName Year Month DayVisits)
38      work.parkOther(keep=ParkCode ParkName);
39     merge work.sorted_code_2016(in=inStats) work.sortedCodes;
40     by ParkCode;
41     if inStats=1 then output work.parkStats;
42     else output work.parkOther;
43 run;
```