```
***********************************************************;
*  Combining Iterative and Conditional DO Loops           *;
***********************************************************;
*  Syntax                                                 *;
*                                                         *;
*    DATA output-table;                                   *;
*       SET input-table;                                  *;
*       . . .                                             *;
*       DO UNTIL | WHILE (expression);                    *;
*          . . . repetitive code . . .                    *;
*          OUTPUT;                                        *;
*       END;                                              *;
*       DO index-column = start TO stop <BY increment>    *;
*          UNTIL | WHILE (expression);                    *;
*          . . . repetitive code . . .                    *;
*          OUTPUT;                                        *;
*       END;                                              *;
*       . . .                                             *;
*       OUTPUT;                                           *;
*    RUN;                                                 *;
***********************************************************;

***********************************************************;
*  Demo                                                   *;
*  1) The intent of both DATA steps is process the DO    *;
*     loop for each row in the PG2.SAVINGS2 table. One    *;
*     DATA step uses DO WHILE and the other uses DO       *;
*     UNTIL. Each loop represents one month of savings.   *;
*     The loop should stop iterating when Savings exceeds *;
*     3000 or 12 months pass, whichever comes first.      *;
*  2) Run the demo program and view the 2 reports that    *;
*     are created. Notice that the values of Savings in   *;
*     the DO WHILE and DO UNTIL reports match, indicating *;
*     that the DO loops executed the same number of times *;
*     for each person.                                    *;
*  3) Observe that for the first row in both the DO WHILE *;
*     and DO UNTIL reports has Month equal to 13. Savings *;
*     did not exceed $5,000 after 12 iterations of the DO *;
*     loop. The Month index variable was incremented to   *;
*     13 at the end of the twelfth iteration of the loop, *;
*     which triggered the end of the loop in both DATA    *;
*     steps and an implicit output action to the output   *;
*     table.                                              *;
*  4) Observe that in rows 2, 3 and 4, the value of Month *;
*     in the DO WHILE results is one greater compared to  *;
*     the DO UNTIL results. This is because in the DO     *;
*     WHILE loop, the index variable Month increments     *;
*     before the condition is checked. Therefore, the     *;
*     Month column in the output data does not accurately *;
*     represent the number of times the DO loop iterated  *;
*     in either DATA step.                                *;
*  5) To create an accurate counter for the number of     *;
*     iterations of a DO loop, make the following         *;
*     modifications to both DATA steps:                   *;
*     a) Add a sum statement inside the loop to create a  *;
*        column named Month and add 1 for each iteration. *;
*     b) Before the DO loop add an assignment statement   *;
*        to reset Month to 0 each time a new row is read  *;
*        from the input table.                            *;
*     c) Change the name of the index variable to an      *;
*        arbitrary name, such as i.                       *;
*     d) Add a DROP statement to drop i from the output   *;
*        table.                                           *;
*  6) Run the program and examine the results. Notice the *;
*     values of Savings and Month match for the DO WHILE  *;
*     and DO UNTIL reports. Month represents the number   *;
*     of times the DO loop executed for each row.         *;
***********************************************************;

data MonthSavingsW;
    set pg2.savings2;
    Month=0;
    do i=1 to 12 while (savings<=5000);
       Month+1;
       Savings+Amount;
       Savings+(Savings*0.02/12);
    end;
    format Savings comma12.2;
    drop i;
run;

data MonthSavingsU;
    set pg2.savings2;
    Month=0;
    do i=1 to 12 until (savings>5000);
       Month+1;
       Savings+Amount;
       Savings+(Savings*0.02/12);
    end;
    format Savings comma12.2;
    drop i;
run;

title "DO WHILE Results";
proc print data=MonthSavingsW;
run;

title "DO UNTIL Results";
```