1. When attempting to minimize memory usage, the most efficient way to do group processing when using the MEANS procedure is to use:

A. the BY statement.
B. GROUPBY with the NOTSORTED specification.
C. the CLASS statement.
D. multiple WHERE statements.

Explanation:

It alludes to the "standard" long-bearded practical trick used in the situation when MEANS/SUMMARY with CLASS uses too much memory. It does not ask what is more efficient in general (which would be to use something else instead of MEANS if possible), just what leads to the minimal memory footprint, and the answer is A.

A:

The reason is that without BY, MEANS creates and builds its AVL tree for the entire file and all categorical values crossings found in CLASS. With BY, the PROC builds the tree for the current BY group only, computes the stats, then erases the tree before the next BY group starts, creates and builds it again, and so on.

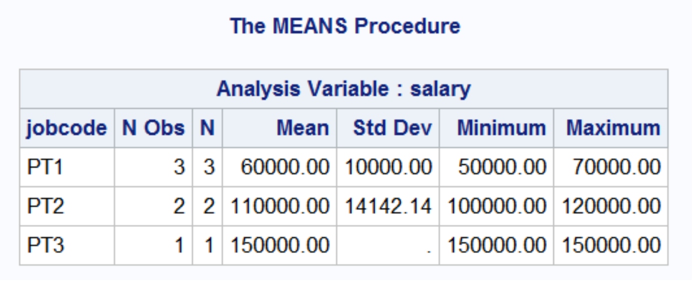
The erase/create process is not a zero cost one, so the smaller the more numerous the BY groups are, the more it slows down the processing; however, the smaller is the largest BY group, the smaller is the memory footprint.

B:

Answer B is just a distraction - a common device in computer-adaptive tests.

Experiments of ‘CLASS’ with SAS:

```
12      proc means data=pilots;
13      class jobcode;
14      run;
NOTE: There were 6 observations read from the data set WORK.PILOTS.
NOTE: PROCEDURE MEANS used (Total process time):
        real time               0.56 seconds
        cpu time                 0.29 seconds
```

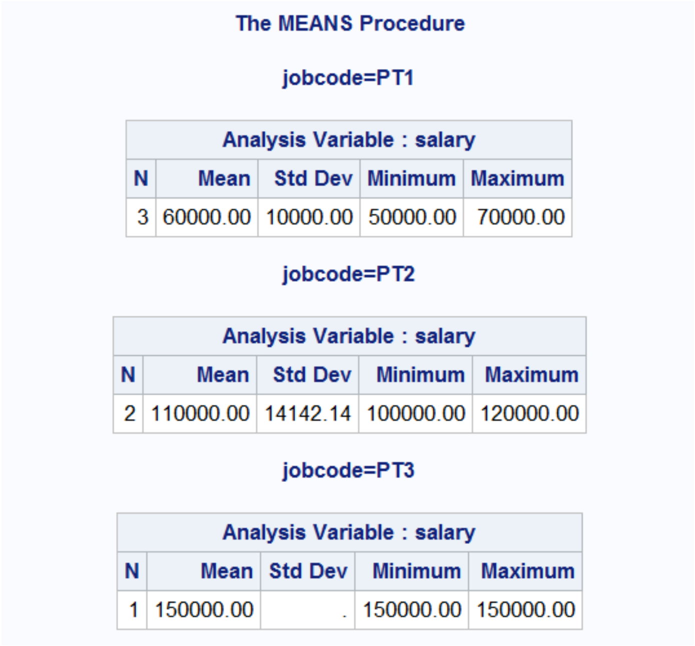


The MEANS Procedure

Analysis Variable : salary						
jobcode	N Obs	N	Mean	Std Dev	Minimum	Maximum
PT1	3	3	60000.00	10000.00	50000.00	70000.00
PT2	2	2	110000.00	14142.14	100000.00	120000.00
PT3	1	1	150000.00	.	150000.00	150000.00

Experiments of ‘BY’ with SAS:

```
-----------------------------------------------------------------------
15      proc means data=pilots;
16      by jobcode;
17      run;
NOTE: There were 6 observations read from the data set WORK.PILOTS.
NOTE: PROCEDURE MEANS used (Total process time):
        real time               0.10 seconds
        cpu time                 0.03 seconds
```



The MEANS Procedure

jobcode=PT1

Analysis Variable : salary				
N	Mean	Std Dev	Minimum	Maximum
3	60000.00	10000.00	50000.00	70000.00

jobcode=PT2

Analysis Variable : salary				
N	Mean	Std Dev	Minimum	Maximum
2	110000.00	14142.14	100000.00	120000.00

jobcode=PT3

Analysis Variable : salary				
N	Mean	Std Dev	Minimum	Maximum
1	150000.00	.	150000.00	150000.00

2. The SAS data set WORK.CHECK has a variable named Id_Code in it. Which SQL statement would create an index on this variable?

A. create index Id_Code on WORK.CHECK;
B. create index(Id_Code) on WORK.CHECK;
C. make index=Id_Code from WORK.CHECK;
D. define index(Id_Code) in WORK.CHECK;

3. Given the SAS data sets:

```
WORK.EMPLOYEE              WORK.NEWEMPLOYEE

  Name        Dept          Names       Salary
  --------    -----        --------    ------
  Alan        Sales         Michelle    50000
  Michelle    Sales         Paresh      60000
```

A SAS program is submitted and
the following is written to the SAS log:

```
101   proc sql;
102      select dept, name
103      from WORK.EMPLOYEE
104      where name=(select names
                    from newemployee
                    where salary > 40000)
ERROR: Subquery evaluated to more than one row.
105   ;
106   quit;
```

What would allow the program to successfully execute without errors?

A. Replace the where clause with:

```
where EMPLOYEE.Name=(select Names delimited with ','
                     from WORK.NEWEMPLOYEE
                     where Salary > 40000);
```

B. Replace line 104 with:

```
where EMPLOYEE.Name =ANY (select Names ~~separated with ','~~
                         from WORK.NEWEMPLOYEE
                         where Salary > 40000);
```

C. Replace the equal sign with the IN operator.

D. Qualify the column names with the table names.

Explanation:

Add “ANY” before the parenthesis or replace the equal sign with the IN operator since the number of values returned by subquery are more than one. If there is only one value returned by the subquery, then the code will be correct.

4. Given the SAS data set SASUSER.HIGHWAY:

```
Steering   Seatbelt   Speed   Status     Count
--------   --------   -----   -------    -----
absent      No          0-29    serious     31
absent      No          0-29    not        1419
absent      No          30-49   serious    191
absent      no         30-49   not        2004
absent      no         50+      serious    216
```

The following SAS program is submitted:

```
proc sql noprint;
      select distinct
            Speed    [_insert_SQL_clause_]
      from SASUSER.HIGHWAY
      ;
quit;

title1 "Speed values represented are: &GROUPS";
proc print data=SASUSER.HIGHWAY;
run;
```

Which SQL clause stores the text **0-29, 30-49, 50+** in the macro variable GROUPS?

A. into &GROUPS

B. into :GROUPS

C. into :GROUPS separated by ','

D. into &GROUPS separated by ','

(2012/11/17 日考题中选项有改动，但是答案没有变。)

5. The SAS data set WORK.CHECK has an index on the variable Code and the following SAS program is submitted.

```
proc sort data=WORK.CHECK;
      by Code;
```

```
  run;
```

Which describes the result of submitting the SAS program?

A. The index on Code is deleted.

B. The index on Code is updated.

C. The index on Code is uneffected.

D. The sort does not execute.

（在于有了 index 的 variable，不能再用 by 来 SORT 了。）

6. The table WORK.PILOTS contains the following data:

```
    WORK.PILOTS
    Id     Name        Jobcode    Salary
   ---    ------     -------    ------
   001    Albert      PT1           50000
   002    Brenda      PT1           70000
   003    Carl        PT1           60000
   004    Donna       PT2           80000
   005    Edward      PT2           90000
   006    Flora       PT3          100000
```

The data set was summarized to include average salary based on jobcode:

```
   Jobcode    Salary        Avg
   -------    ------      -----
   PT1          50000     60000
   PT1          70000     60000
   PT1          60000     60000
   PT2          80000     85000
   PT2          90000     85000
   PT3         100000    100000
```

Which SQL statement could NOT generate this result?

```
A. select
   Jobcode,
   Salary,
   avg(Salary) label='Avg'
from WORK.PILOTS
group by Jobcode
order by Id
;
```

```
B. select
   Jobcode,
   Salary,
   (select avg(Salary)
   from WORK.PILOTS as P1
   where P1.Jobcode=P2.Jobcode) as Avg
from WORK.PILOTS as P2
order by Id
;
C. select
   Jobcode,
   Salary,
   (select avg(Salary)
   from WORK.PILOTS
   group by Jobcode) as Avg
from WORK.PILOTS
order by Id
;
D. select
   Jobcode,
   Salary,
   Avg
from
   WORK.PILOTS,
  (select
      Jobcode as Jc,
      avg(Salary) as Avg
   from WORK.PILOTS
   group by 1)
where Jobcode=Jc
order by Id
;
```

(define variable Avg in subquery in from clause.)

Explanation:

B: 这里利用的是自身的数据集，分别设置不同的别名，可以当做两个数据集来使用。不过这里 WHERE 语句很值得学习，我个人对这里的 WHERE 语句这样理解的：(select avg(Salary) from WORK.PILOTS as P1 where P1.Jobcode=P2.Jobcode)启示执行之后返回的是三个 AVG 均值，但是因为这三个 AVG 均值都是与 JOBCODE 匹配的，那么前面的 select Jobcode, Salary，所以当执行到(select avg(Salary) from WORK.PILOTS as P1 where P1.Jobcode=P2.Jobcode)这里时，实际上是选择了 select jobcode 中 jobcode 对应的 avg 均值。

C: 子查询涉及不止一项。

7. A quick rule of thumb for the space required to run PROC SORT is:

A. two times the size of the SAS data set being sorted.

B. three times the size of the SAS data set being sorted.

C. four times the size of the SAS data set being sorted.

D. five times the size of the SAS data set being sorted.

Explanation

(140-2010: Dear Miss SASAnswers: A Guide to Sorting Your Data 中的描述是这样的:

If you want the sort to complete entirely in memory, a simple rule of thumb is four times the size of the data set.

In releases prior to SAS 9, the required workspace is approximately three to four times the size of the data file. Beginning with SAS 9, the required workspace is approximately twice the size of the data file. The workspace can be allocated in memory and/or on disk as a utility file, depending on which sort utility and options are specified.

If you want the sort to complete entirely in memory, a simple rule of thumb is four times the size of the data set.

And I'm assuming the data set is not compressed or being subset with a DROP=/KEEP= data set option or a WHERE statement.

A better estimate would be to use this formula to predict the amount of memory:

((length of observation + sum of lengths of BY variables)*number of observations)* **1.10**

The amount of space that the SAS sort needs depends on the following conditions:

- whether the sort can be done with threading
- the length of the observations
- the number of variables in the BY statement and their storage lengths
- the operating environment in which PROC SORT executes
- whether the LINGUISTIC= option is being used (that takes more memory)

Use the SORTSIZE= option in the PROC SORT statement to do the following:

- specify the amount of memory that is available to the SORT procedure
- improve the sort performance by restricting the swapping of memory to disk that is controlled by the operating system

8. Multi-threaded processing for PROC SORT will affect which of these system resources?

A. CPU time will decrease, wall clock time will decrease

B. CPU time will increase, wall clock time will decrease

C. CPU time will decrease, wall clock time will increase

D. CPU time will increase, wall clock time will increase

9. Given the SAS data set WORK.TRANSACT:

Rep	Cost	Ship
-----	-----	-----
SMITH	200	50
SMITH	400	20
JONES	100	10

```
      SMITH           600       100
      JONES           100         5
```

The following output is desired:

```
      Rep
      -----     ----
      JONES       105
      SMITH       250
```

Which SQL statement was used?

```
 A. select
      rep,
      min(Cost+Ship)
from WORK.TRANSACT
order by Rep
;
```

```
B. select
      Rep,
      min(Cost,Ship) as Min
from WORK.TRANSACT
summary by Rep
order by Rep
;
```

```
C. select
      Rep,
      min(Cost,Ship)
from WORK.TRANSACT
group by Rep
order by Rep
;
```

（改成 min(sum(cost,ship))才对）

```
D. select
      Rep,
      min(Cost+Ship)
from WORK.TRANSACT
group by Rep
order by Rep
;
```

10. (%EVAL) The following SAS program is submitted:

```
%let Value=9;
%let Add=5;
%let Newval=%eval(&Value/&Add);
%put &Newval;
```

What is the value of the macro variable Newval when the %PUT statement executes?

A. 0.555

B. 2

C. 1.8

D. 1

Explanation:

The %EVAL function evaluates integer arithmetic or logical expressions. Logical expressions and arithmetic expressions are sequences of operators and operands forming sets of instructions that are evaluated to produce a result.

- An arithmetic expression contains an arithmetic operator.
- A logical expression contains a logical operator.

General form, %EVAL function:

```
%EVAL (arithmetic or logical expression);
```

The %EVAL function

- translates integer strings and hexadecimal strings to integers
- translates tokens representing arithmetic, comparison, and logical operators to macro-level operators
- performs arithmetic and logical operations.

For arithmetic expressions, if an operation results in a non-integer value, %EVAL truncates the value to an integer. **Also, %EVAL returns a null value and issues an error message when non-integer values are used in arithmetic expressions.**

%EVAL evaluates logical expressions and returns a value to indicate if the expression is true or false. A value of 0 indicates that the expression is false, and a value of 1 or any other numeric value indicates that the expression is true.

The %EVAL function does not convert the following to numeric values:

- numeric strings that contain a period or E-notation
- SAS date and time constants.

11. The following SAS code is submitted:

```
   data WORK.TEMP WORK.ERRORS / view=WORK.TEMP;
      infile RAWDATA;
      input Xa Xb Xc;
      if Xa=. then output WORK.ERRORS;
      else output WORK.TEMP;
  run;
```

Which of the following is true of the WORK.ERRORS data set?

A. The data set is created when the DATA step is submitted.

B. The data set is created when the view TEMP is used in another SAS step.

C. The data set is not created because the DATA statement contains a syntax error.

D. The descriptor portion of WORK.ERRORS is created when the DATA step is submitted.

Explanation:

SAS create these data files when the view is processed in a subsequent DATA/PROC step.

If no such step, then these files will not be created. (So-called partially compiled in DATA view step).

A DATA step view cannot contain global statements, host-specific data set options, or most host-specific **FILE and INFILE** statements. Also, a DATA step view cannot be indexed or compressed.

假设在 d 盘建 SAS 文件夹下有 view.txt 文件：

```
.   10 11
10 11 12
.   10 12
21 22 23
```

在 SAS 编程如下：

```
filename rawdata 'd:\sas\view.txt';
data WORK.TEMP WORK.ERRORS / view=WORK.TEMP;
infile rawdata;
input Xa Xb Xc;
      if Xa=. then output WORK.ERRORS;
      else output WORK.TEMP;
run;

proc print data=temp;
run;
```

WORK.TEMP is created, by not WORK.ERROR. (check with CONTENTS PROC.)

12. Which title statement would always display the current date?

A. title "Today is: &sysdate.";
B. title "Today is: &sysdate9.";
C. title "Today is: &today.";
D. title "Today is: %sysfunc(today(),worddate.)";

Explanation:

A and B show the time SAS start. (A: 07Apr14; B: 07Apr2014.)

If C is corrected as: title "Today is: %sysfunc(today())."; it shows a SAS date number.

D: April 7, 2014.

13. Given the SAS data sets:

```
WORK.ONE                    WORK.TWO

Id      Name             Id         Salary
---     ------           ---        ------
112     Smith            243        150000
243     Wei              355         45000
457     Jones            523         75000
```

The following SAS program is submitted:

```
data WORK.COMBINE;
merge WORK.ONE WORK.TWO;
by Id;
run;
```

(like horizontal full join)

Which SQL procedure statement produces the same results?

A. create table WORK.COMBINE as

```
select
   Id,
   Name,
   Salary
from
   WORK.ONE
   full join
   WORK.TWO
on ONE.Id=TWO.Id
;
```

B. create table WORK.COMBINE as

```
select
     coalesce(ONE.Id, TWO.Id) as Id,
     Name,
     Salary
from
     WORK.ONE,
     WORK.TWO
where ONE.Id=TWO.Id
;
```

```
C. create table WORK.COMBINE as
select
     coalesce(ONE.Id, TWO.Id) as Id,
     Name,
     Salary
from
     WORK.ONE
     full join
     WORK.TWO
on ONE.Id=TWO.Id
order by Id
;
```

```
D. create table WORK.COMBINE as
select
     coalesce(ONE.Id, TWO.Id) as Id,
     Name,
     Salary
from
     WORK.ONE,
     WORK.TWO
where ONE.Id=TWO.Id
order by ONE.Id
;
```

Explanation:

DATA step match-merges and PROC SQL joins can produce the same results, although there are important differences between these two techniques.

- When all the values of the BY variable (column) match and there are no duplicate BY variables, you can use a PROC SQL inner join.
- When only some of the values of the BY variable match, you can use a PROC SQL full outer join. To overlay common columns, you must use the COALESCE function with the PROC SQL join.

(Chapter 3: When Only Some of the Values Match: Using the COALESCE Function)

A: Error message: Ambiguous reference, column Id is in more than one table (in the second line).

B and D: Table WORK.COMBINE created, with 1 rows and 3 columns because of the where statement.

14. The following SAS program is submitted:

```
proc contents data=TESTDATA.ONE;
run;
```

Which SQL procedure step produces similar information about the column attributes of TESTDATA.ONE?

```
A. proc sql;
      contents from TESTDATA.ONE;
quit;
B. proc sql;
      describe from TESTDATA.ONE;
quit;
C. proc sql;
      contents table TESTDATA.ONE;
quit;
D. proc sql;
      describe table TESTDATA.ONE;
quit;
```

15. Given the SAS data set WORK.ONE:

```
  Rep         Cost
 -----       ----
 SMITH         200
 SMITH         400
 JONES        100
 SMITH         600
 JONES        100
```

The following SAS program is submitted;

```
proc sql;
    select
        Rep,
        avg(Cost)
    from WORK.ONE
    order by Rep
    ;
quit;
```

Which result set would be generated?

A.
```
JONES       280
JONES       280
SMITH       280
SMITH       280
SMITH       280
```
B.
```
JONES       600
SMITH       100
```
C.
```
JONES       280
SMITH       280
```
D.
```
JONES       100
JONES       100
SMITH       600
SMITH       600
SMITH       600
```

16. Given the SAS data sets:

```
   WORK.MATH1A                 WORK.MATH1B
   Name        Fi            Name       Fi
 -------    --              ------    --
 Lauren      L               Smith      M
 Patel       A              Lauren     L
 Chang       Z               Patel      A
 Hillier    R
```

The following SAS program is submitted:

```
   proc sql;
      select *
      from WORK.MATH1A
      [_insert_set_operator_]
      select *
      from WORK.MATH1B
      ;
 quit;
```

The following output is desired:

```
Name        Fi
-------    --
Lauren      L
Patel       A
Chang        Z
Hillier    R
Smith       M
Lauren      L
Patel       A
```

Which SQL set operator completes the program and generates the desired output?

A. append corr

B. union corr

C. outer union corr

D. intersect corr

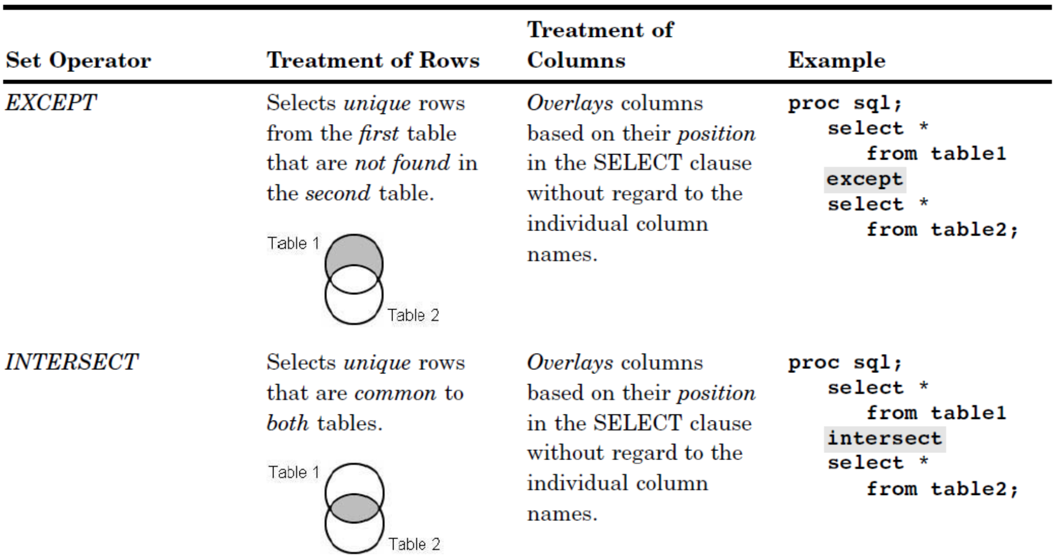
Explanation:

A: Not a valid option;

B: Delete duplicated rows (select unique rows); removes nonmatching columns.

C: Keep duplicate rows; overlays same-named columns and displays columns that have nonmatching names without overlaying (using corr to overlay by the same names).

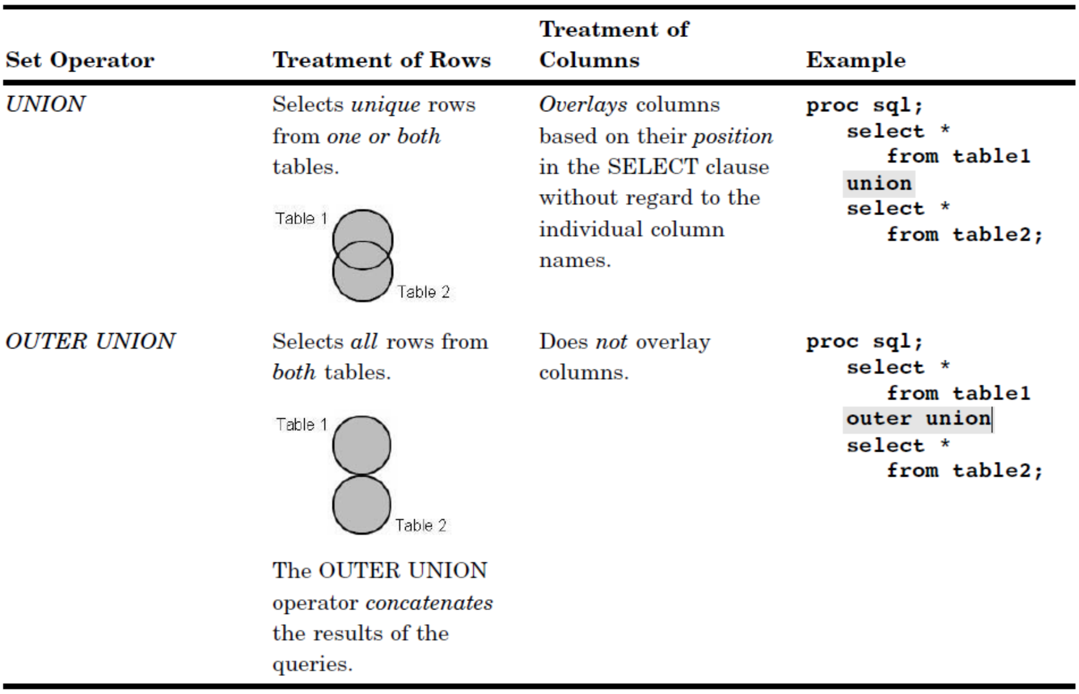
D: Keep matched rows, and removes nonmatching columns.



Set Operator	Treatment of Rows	Treatment of Columns	Example

EXCEPT	Selects *unique* rows from the *first* table that are *not found* in the *second* table.	*Overlays* columns based on their *position* in the SELECT clause without regard to the individual column names.	`proc sql;`
			`select *`
			`from table1`
			`except`
			`select *`
			`from table2;`

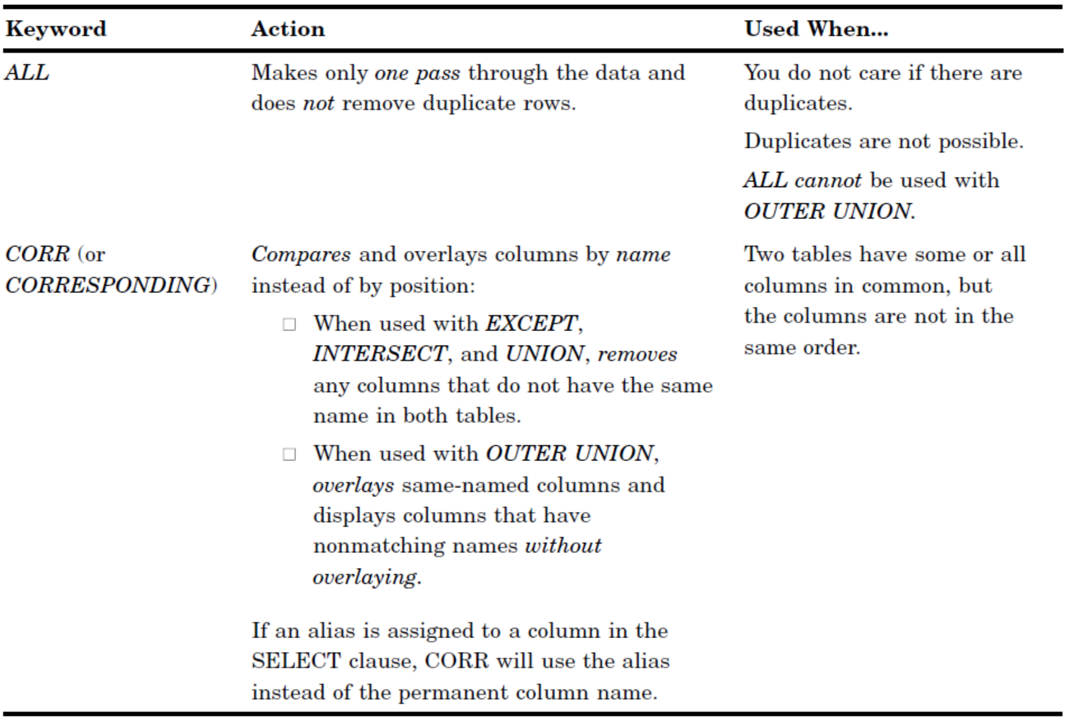
INTERSECT	Selects *unique* rows that are *common* to *both* tables.	*Overlays* columns based on their *position* in the SELECT clause without regard to the individual column names.	`proc sql;`
			`select *`
			`from table1`
			`intersect`
			`select *`
			`from table2;`



Set Operator	Treatment of Rows	Treatment of Columns	Example

UNION	Selects *unique* rows from *one or both* tables.
	Table 1
	Table 2
		Overlays columns based on their *position* in the SELECT clause without regard to the individual column names.	`proc sql;`
			`   select *`
			`      from table1`
			`   union`
			`   select *`
			`      from table2;`

OUTER UNION	Selects *all* rows from *both* tables.
	Table 1
	Table 2
	The OUTER UNION operator *concatenates* the results of the queries.
		Does *not* overlay columns.	`proc sql;`
			`   select *`
			`      from table1`
			`   outer union`
			`   select *`
			`      from table2;`



Keyword	Action	Used When...
ALL	Makes only *one pass* through the data and does *not* remove duplicate rows.	You do not care if there are duplicates. Duplicates are not possible. *ALL cannot* be used with *OUTER UNION*.
CORR (or *CORRESPONDING*)	*Compares* and overlays columns by *name* instead of by position: □ When used with *EXCEPT*, *INTERSECT*, and *UNION*, *removes* any columns that do not have the same name in both tables. □ When used with *OUTER UNION*, *overlays* same-named columns and displays columns that have nonmatching names *without overlaying*. If an alias is assigned to a column in the SELECT clause, CORR will use the alias instead of the permanent column name.	Two tables have some or all columns in common, but the columns are not in the same order.

17. Which of the following is an advantage of SAS views?

A. SAS views can access the most current data in files that are frequently updated.

B. SAS views can avoid storing a SAS copy of a large data file.

C. SAS views can decrease programming time.

D. both A and B are true

18. In what order does SAS search for format definitions by default?

A. 1. WORK.FORMATS 2. LIBRARY.FORMATS

B. 1. LIBRARY.FORMATS 2. WORK.FORMATS

C. There is no default order, it must be defined by the user.

D. All user defined libraries that have a catalog named FORMATS, in alphabetic order.

19. Function %UPCASE:

Given the dataset WORK.STUDENTS:

```
  Name        Age
  -------     ---
  Mary         15
  Philip      16
  Robert      12
  Ronald       15
```

The following SAS program is submitted:

```
    %let Value=Philip;
  proc print data=WORK.STUDENTS;
       [_insert_WHERE_statement_]
  run;
```

Which WHERE statement successfully completes the program and produces a report?

A. where upcase(Name)=upcase(&Value);

B. where upcase(Name)=%upcase(&Value);

C. where upcase(Name)="upcase(&Value)";

D. where upcase(Name)="%upcase(&Value)";

Explanation:

The %UPCASE function enables you to change the value of a <u>macro variable</u> from lowercase to uppercase before substituting that value in a SAS program. Quotation marks are needed since the both sides of equal sign are character strings.

20. The following SAS program is submitted:

```
   data WORK.TEMP;
      length A B 3 X;
      infile RAWDATA;
      input A B X;
 run;
```

What is the length of variable A?

A. 3

B. 8

C. WORK.TEMP is not created - X has an invalid length.

D. Unknown.

Explanation:

A: 这里考察的是 LENGTH 语句对变量长度的约束。

如果没有 Length 语句的约束，如 X 就是默认的长度了 8 个 BYTE。

LENGTH 语句制定了 A B 的长度，但是没有指定 X 的长度，如果 X 长度不需要 LENGTH 特别指出那么就不用 LENGTH X 这样的语句，因为 LOG 里面提示你没有设置长度值，不符合 LENGTH 语句语法。

这里需要指出的是，程序编译时就已经设置了长度，要么是默认的，要么是指定的。在这里也许数据集 WORK.TEMP 没有观测；但是通过 CONTENTS 依然能看到每一个变量的属性。

Warning message from SAS log:

WARNING: The data set WORK.TEMP may be incomplete. When this step was stopped there were 0 observations and 3 variables.

D: Using PROC DATASETS/CONTENTS, we can see that X has a length of 8 bytes.

21. The following SAS program is submitted:

```
data WORK.NEW;
     do i=1, 2, 3;
          Next=cats('March' || i );
          infile XYZ
               filevar=Next
               end=Eof;
          do until (Eof);
               input Dept $ Sales;
          end;
     end;
 run;
```

The purpose of the FILEVAR=option on the INFILE statement is to name the variable Next, whose value:

A. points to a new input file.

B. is output to the SAS data set WORK.NEW.

C. is an input SAS data set reference.
D. points to an aggregate storage location.

Explanation:

A: You can make the process of concatenating raw data files more flexible by using an INFILE statement with the FILEVAR=option. The FILEVAR=option enables you to dynamically change the currently opened input file to a new input file.

General form

INFILE file-specification FILEVAR= variable;

where

FILEVAR= *variable*

names a variable whose change in value causes the INFILE statement to close the current input file and open a new input file.

variable

contains a character string that is a physical filename.

```
data WORK.NEW;
    do i=1, 2, 3;
      Next=cats('March' || i ); *Three data sets in library WORK: March1, March2, March3
      infile XYZ                *XYZ is a space holder.
        filevar=Next            *Use defined file name defined above
        end=Eof;
      do until (Eof);
         input Dept $ Sales;
      end;
    end;
  run;
```

end=Eof; When you read past the last record in a raw data file, the DATA step normally stops processing. In this case, you need to read the last record in the first two raw data files. However, you do not want to read past the last record in either of those files because the DATA step will stop processing. You can use the END= option with the INFILE statement to determine when you are reading the last record in the last raw data file.

C: it’s SET statement options.

22. DESCRIBE TABLE Statement

Given the following partial SAS log:

```
NOTE: SQL table SASHELP.CLASS was created like:
create table SASHELP.CLASS( bufsize=4096 )
  (
    Name char(8),
    Sex char(1),
    Age num,
    Height num,
    Weight num
  );
```

Which SQL procedure statement generated this output?

A. CONTENTS FROM SASHELP.CLASS;

B. CREATE FROM SASHELP.CLASS INTO LOG;

C. DESCRIBE TABLE SASHELP.CLASS;

D. VALIDATE SELECT * FROM SASHELP.CLASS;

Explanation:

A: under PROC DATASETS.

23. Given the SAS data set SASUSER.HIGHWAY:

Steering	Seatbelt	Speed	Status	Count
--------	--------	-----	-------	-----
absent	No	0-29	serious	31
absent	No	0-29	not	1419
absent	No	30-49	serious	191
absent	no	30-49	not	2004
absent	no	50+	serious	216

The following SAS program is submitted:

```
%macro SPLIT;
 proc sort data=SASUSER.HIGHWAY
            out=WORK.UNIQUES(keep=Status)   nodupkey;
      by Status;
 run;

 data _null_;
      set uniques end=Lastobs;
      call symputx('Status'||left(_n_),Status);
      if Lastobs then call symputx('Count',_n_);
 run;

 %local i;
  data
      %do i=1 %to &count;
         [_insert_reference_]
      %end;
      ;
      set SASUSER.HIGHWAY;
      select (Status);
         %do i=1 %to &Count;
            when("[_insert_reference_]") output [_insert_reference_];
         %end;
         otherwise;
      end;
   run;
 %mend;
  %SPLIT
```

What macro variable reference completes the program to create the WORK.NOT and WORK.SERIOUS data sets?

A. &Status&i

B. &&Status&i

C. &Status&Count
D. &&Status&Count

Explanation:

OPTION NODUPKEY in PROC SORT: Remove duplicate observations.
Function CALL SYMPUTX (*Macro-variable*, *value*): Assign *value* to *macro variable* under DATA PROC; at the meanwhile, remove any leading or trailing blanks. Function CALL SYMPUTX doesn't remove any blanks.
CALL SYMPUTX ('Status'||left(_n_),Status); Assign the value of Status to macro variable Status***1***-Status ***n***

24. The following SAS program is submitted:

```
%let Num1=7;
%let Num2=3;
%let Result=%eval(&Num1/&Num2);
%put &Result;
```

What is the value of the macro variable Result when the %PUT statement executes?
A. 2.3
B. 2
C. . (missing value)
D. 2.33333333333333

25. Macros That Include Keyword Parameters:

Given the SAS data set SASUSER.HIGHWAY:

```
Steering Seatbelt Speed Status    Count
-------- -------- ----- ------- -----
absent    No        0-29   serious    31
absent    No        0-29   not      1419
absent    No        30-49 serious    191
absent    no        30-49 not       2004
absent    no        50+    serious    216
```

The following SAS program is submitted:

```
%macro HIGHWAY(Belt=no);
   proc print data=SASUSER.HIGHWAY;
      where Seatbelt="&Belt" ;
   run;
%mend;

%HIGHWAY(Belt=No)
```

How many observations appear in the generated report?

A. 0

B. 2

C. 3

D. 5

Explanation:

Macros That Include Keyword Parameters

Like positional parameters, keyword parameters create macro variables. However, when you use keyword parameters to create macro variables, you list both the name and the value of each macro variable in the macro definition.

Keyword parameters can be listed in any order. Whatever value you assign to each parameter (or variable) in the %MACRO statement becomes its default value. Null values are allowed.

C since WHERE clause is case sensitive

26. Given the following SAS data sets:

```
   WORK.VISIT1                   WORK.VISIT2

 Id       Expense              Id       Cost
 ---      -------             ---      ----
 001          500              001        300
 001          400              002        600
 003          350
```

The following result set was summarized and consolidated using the SQL procedure:

```
 Id       Cost
 ---      ----
 001        300
 001        900
 002        600
 003        350
```

Which of the following SQL statements was most likely used to generate this result?

```
A. select
   Id,
   sum(Expense) label='COST'
from WORK.VISIT1
group by 1
union all
select
   Id,
   sum(Cost)
```

```
from WORK.VISIT2
group by 1
order by 1,2
;
```

B.

```
 select
    id,
    sum(expense) as COST
from
    WORK.VISIT1(rename=(Expense=Cost)),
    WORK.VISIT2
where VISIT1.Id=VISIT2.Id
group by Id
order by Id,Cost
;
```

C.

```
 select
    VISIT1.Id,
    sum(Cost) as Cost
from
    WORK.VISIT1(rename=(Expense=Cost)),
    WORK.VISIT2
where VISIT1.Id=VISIT2.Id
group by Id
order by Id,Cost
;
```

D.

```
 select
    Id,
    sum(Expense) as Cost
from WORK.VISIT1
group by Id
outer union corr
select
    Id,
    sum(Cost)
from WORK.VISIT2
group by Id
order by 1,2
;
```

Explanation:

A: When the keyword ALL is added to the UNION operator, the output displays all rows from both tables, both unique and duplicate. With followed CORR, only matched column will be kept.

B: Was renamed by below rename statement;

B and C，in clause of “ORDER BY”, didn’t specify which data set for variable ID.

D has three columns, since “OUTER UNION CORR” does not remove un-matching column: Expense.

27. Given the SAS data sets:

```
    WORK.FIRST              WORK.SECOND
   Common    X              Common    Y
  ------   --             ------   --
  A        10               A        1
  A        13               A        3
  A        14               B        4
  B         9               B        2
```

The following SAS program is submitted:

```
 data WORK.COMBINE;
     set WORK.FIRST;
     set WORK.SECOND;
  run;
```

What data values are stored in data set WORK.COMBINE?

```
A.  Common    X    Y
   ------   --   --
   A        10    1
   A        13    3
   B        14    4
   B         9    2
B.Common    X    Y
  ------   --   --
  A        10    1
  A        13    3
  A        14    3
  B         9    4
  B         9    2
C.Common    X    Y
  ------   --   --
  A        10    1
  A        13    3
  A        14    .
  B         9    4
  B         .   2
D.Common   X    Y
```

```
------    --   --
A         10    1
A         13    1
A         14    1
A         10    3
A         13    3
A         14    3
B          9    4
B          9    2
```

28. Which of the following ARRAY statements is similar to the statement array Yr{1974:2007} Yr1974-Yr2007; and will compile without errors?

A. array Yr{34} Yr1974-Yr2007;
B. array Yr{74:07} Yr1974-Yr2007;
C. array Yr{74-07} Yr1974-Yr2007;
D. array Yr{1974-2007} Yr1974-Yr2007;

29. The following program is submitted to check the variables Xa, Xb, and Xc in the SASUSER.LOOK data set:

```
data _null_ WORK.BAD_DATA / view=WORK.BAD_DATA ;
    set SASUSER.LOOK(keep=Xa Xb Xc);
    length _Check_ $ 10 ;
    if Xa=. then _check_=trim(_Check_)!!" Xa" ;
    if Xb=. then _check_=trim(_Check_)!!" Xb" ;
    if Xc=. then _check_=trim(_Check_)!!" Xc" ;
    put Xa= Xb= Xc= _check_= ;
run ;
```

When is the PUT statement executed?
A. when the code is submitted
B. only when the WORK.BAD_DATA view is used
C. both when the code is submitted and the view is used
D. never, the use of _null_ in a view is a syntax error

30. The following SAS program is submitted:

```
%let product=merchandise;
[_insert_%put_statement_]
```

and the following message is written to the SAS log: the value is "merchandise"

Which macro statement wrote this message?
A. %put the value is '"'&product.'"';
B. %put the value is %quote(&product.);
C. %put the value is "&product.";

D. %put the value is ""&product."";

Explanation:

B should be corrected as %sysfunc(quote(&product.))

31. Given the SAS data sets:

```
   WORK.ONE              WORK.TWO

   X          Y          SumY
   --       --           ----
   A         10            36
   A          3
   A         14
   B          9
```

The following SAS DATA step is submitted:

```
data WORK.COMBINE;
     if _n_=1 then set WORK.TWO;
     set WORK.ONE;
run;
```

What data values are stored in data set WORK.COMBINE?

A. An ERROR message is written to the SAS log and the data set WORK.COMBINE is not created.

```
B. SumY      X      Y
   ----     --     --
     36      A     10
```

```
C.   SumY      X      Y
   ----     --     --
     36      A     10
      .      A      3
      .      A     14
      .      B      9
```

```
D. SumY      X      Y
   ----     --     --
     36      A     10
     36      A      3
     36      A     14
     36      B      9
```

Explanation:

内部控制变量_n_表示数据集中的第 1 行观测值。

(一般来说，在读取外部数据生成 SAS 数据集时，系统在数据部迭代开始的时候为每一个变量设

置为缺失值，但是一般下列情况不会受此限制：

1.retain 语句中的变量

2.sum 语句中创建的变量

3.数据_temporary_中的对象

4.FILE 和 INFILE 中创建的变量

5.自动变量)

32. The following SAS program is submitted:

```
   data WORK.NEW(bufno=4);
      set WORK.OLD(bufno=3);
  run;
```

Why are the BUFNO options used?

A. to reduce memory usage

B. to reduce CPU time usage

C. to reduce the amount of data read

D. to reduce the number of I/O operations

33. Given the following program and desired results:

```
  %let Thing1=gift;
  %let Thing2=surprise;
  %let Gift1=book;
  %let Gift2=jewelry;
  %let Surprise1=dinner;
  %let Surprise2=movie;

  %let Pick=2;
  %let Choice=surprise;
```

Desired %PUT Results in LOG:

```
  My favorite surprise is a movie
```

What is the correct %PUT statement that generates the desired results?

A. %put My favorite &Thing&Pick is a &&Choice&Pick;

B. %put My favorite &&Thing&pick is a &&&Choice&Pick;

C. %put My favorite &Choice&pick is a &&Thing&Pick;

D. %put My favorite &&Choice&pick is a &&&Thing&Pick;

Explanation:

记住一个原则就是“&&”=“&”

B 中，&&Thing&pick=&+thing+&pick=&+thing+2=&thing2=surprise

&&&Choice&Pick=&+&choice+&pick=&+surprise+2=&surprise2=movie.

当然这样的加法表述不是很严谨，但是很容易理解.

34. Given the SAS dataset WORK.ONE

```
Name        Salary
-----       ------
Hans           200
Maria          205
Jose           310
Ariel          523
```

The following SAS program is submitted:

```
proc sql;
    [_insert_select_clause_]
    from WORK.ONE
    ;
quit;
```

The following output is desired:

```
Salary          Bonus
------          -----
   200             20
   205           20.5
   310             31
   523           52.3
```

Which SQL procedure clause completes the program and generates the desired output?

A. select Salary   Bonus as Salary*.10

B. select Salary   Bonus=Salary*.10 'Bonus'

C. select Salary, Salary*.10 label='Bonus'

D. select Salary, Salary*.10 column="Bonus"

35. The following SAS program is submitted:

```
options reuse=YES;
data SASUSER.REALESTATE(compress=CHAR);
    set SASUSER.HOUSES;
run;
```

What is the effect of the reuse=YES SAS system option?

A. It allows updates in place.

B. It tracks and recycles free space.

C. It allows a permanently stored SAS data set to be replaced.

D. It allows users to access the same SAS data set concurrently.

36. Which statement is true for Data step HASH objects?

A. The key component must be numeric.

B. The data component may consist of numeric and character values.

C. The HASH object is created in one step and referenced in another.

D. The HASH object must be smaller than 2 to the 8th power bytes.

Explanation:

The data component may consist of numeric and character values. And it could be created and then referenced in one step.

37. Given the SAS data sets:

```
   WORK.CLASS1              WORK.CLASS2
   Name     Course          Name     Class
  ------   ------          -------  -----
  Lauren   MATH1           Smith    MATH2
  Patel    MATH1           Farmer   MATH2
  Chang    MATH1           Patel    MATH2
  Chang    MATH3           Hillier  MATH2
```

The following SAS program is submitted:

```
proc sql;
   select Name
   from WORK.CLASS1
   [_insert_set_operator_]
   select Name
   from WORK.CLASS2
   ;
quit;
```

The following output is desired:

```
      Name
      ------
      Chang
      Chang
      Lauren
```

Which SQL set operator completes the program and generates the desired output?

A. intersect corr

B. except all

C. intersect all

D. left except

38. The following SAS program is submitted:

```
    %macro CHECK(Num=4);
        %let Result=%eval(&Num gt 5);
        %put Result is &result;
    %mend;
    %check(Num=10)
```

What is written to the SAS log?

A. Result is 0

B. Result is 1

C. Result is 10 gt 5

D. Result is true

39. The following SAS program is submitted:

```
%let Mv=shoes;
%macro PRODUCT(Mv=bicycles);
    %let Mv=clothes;
%mend;

%PRODUCT(Mv=tents)
%put Mv is &Mv;
```

What is written to the SAS log?

A. Mv is bicycles

B. Mv is clothes

C. Mv is shoes

D. Mv is tents

40. Which of the following SAS System options can aid in benchmarking?

A. BUFSIZE= and BUFNO=

B. FULLSTIMER

C. IOBLOCKSIZE=

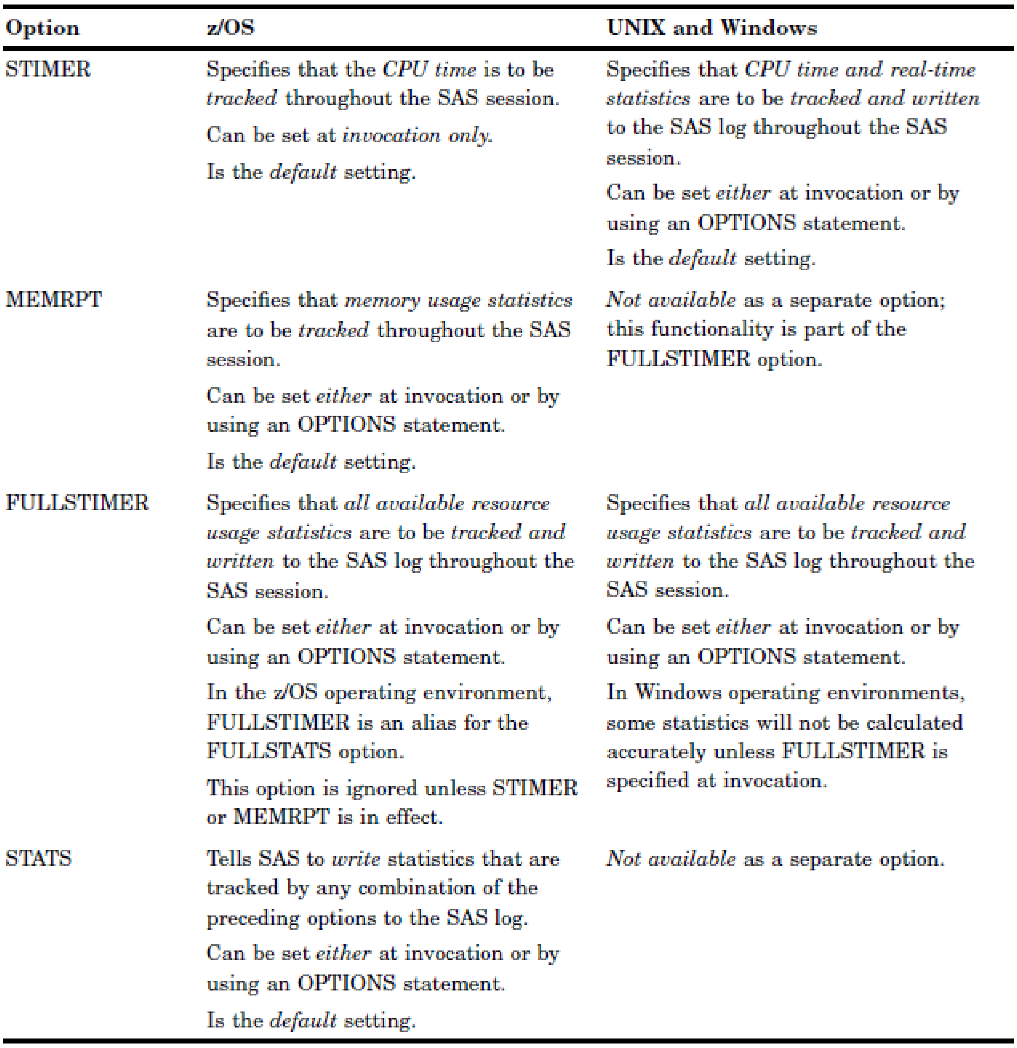
D. SYSTIMER

Explanation:

The PROC SQL option STIMER | NOSTIMER specifies whether PROC SQL writes timing information for each statement to the SAS log, instead of writing a cumulative value for the entire procedure. NOSTIMER is the default.

The STIMER option in PROC SQL is useful when an operation can be accomplished in more than one way and you are benchmarking each technique. Although factors such as code readability and maintenance come into consideration, you might also want to know which PROC SQL step runs the fastest.

You can specify one or more of the SAS system options STIMER, MEMRPT, FULLSTIMER, and STATS to track and report on resource utilization.



Option	z/OS	UNIX and Windows
STIMER	Specifies that the *CPU time* is to be *tracked* throughout the SAS session. Can be set at *invocation only*. Is the *default* setting.	Specifies that *CPU time and real-time statistics* are to be *tracked and written* to the SAS log throughout the SAS session. Can be set *either* at invocation or by using an OPTIONS statement. Is the *default* setting.
MEMRPT	Specifies that *memory usage statistics* are to be *tracked* throughout the SAS session. Can be set *either* at invocation or by using an OPTIONS statement. Is the *default* setting.	*Not available* as a separate option; this functionality is part of the FULLSTIMER option.
FULLSTIMER	Specifies that *all available resource usage statistics* are to be *tracked and written* to the SAS log throughout the SAS session. Can be set *either* at invocation or by using an OPTIONS statement. In the z/OS operating environment, FULLSTIMER is an alias for the FULLSTATS option. This option is ignored unless STIMER or MEMRPT is in effect.	Specifies that *all available resource usage statistics* are to be *tracked and written* to the SAS log throughout the SAS session. Can be set *either* at invocation or by using an OPTIONS statement. In Windows operating environments, some statistics will not be calculated accurately unless FULLSTIMER is specified at invocation.
STATS	Tells SAS to *write* statistics that are tracked by any combination of the preceding options to the SAS log. Can be set *either* at invocation or by using an OPTIONS statement. Is the *default* setting.	*Not available* as a separate option.

41. Given the following macro program:

```
   %macro MAKEPGM(NEWNAME, SETNAME, PRINT);
      data &NEWNAME;
          set &SETNAME;
      run;
      %if &PRINT=YES %then %do;
          proc print data=&NEWNAME.(obs=10);
          run ;
      %end;
  %mend;
```

Which option would provide feedback in the log about the parameter values passed into this macro when invoked?

A. MPRINT

B. MDEBUG

C. MLOGIC

D. MPARAM

Explanation:

A: it just prints the SAS statements in the macros.

C: it can give the messages about what parameter is resolved and used in certain step.

42. The NOTSORTED option on the BY statement cannot be used with which other statement or option?

A. SET

B. MERGE

C. IF FIRST.by-variable

D. BY GROUPFORMAT by-variable

Explanation:

NOTSORTED option on the BY statement cannot be used with statements of MERGE and UPDATE Statement.

The NOTSORTED option specifies that observations that have the same BY value are grouped together but are not necessarily sorted in alphabetical or numeric order.

The NOTSORTED option can appear anywhere in the BY statement and is useful if you have data that is in logical categories or groupings such as chronological order. The NOTSORTED option works best when observations that have the same BY value are stored together.

CAUTION:

The NOTSORTED option turns off sequence checking. If your data is not grouped, using the NOTSORTED option can produce a large amount of output.

CAUTION:

The NOTSORTED option cannot be used with the MERGE or UPDATE statements.

A BY statement does not use an index if the BY statement includes the DESCENDING or NOTSORTED option or if SAS detects that the data file is physically stored in sorted order on the BY variables.

43. Given the SAS data set WORK.ONE:

```
  Rep        Cost
-----      ----
SMITH        200
SMITH        400
JONES        100
SMITH        600
JONES        100
```

The following SAS program is submitted:

```
proc sql;
    select
          Rep,
          avg(Cost) as Average
    from WORK.ONE
    [either__insert_SQL_where_clause_]
    group by Rep
    [_or_    _insert_SQL_having_clause_]
    ;
quit;
```

The following output is desired:

```
Rep              Average
-----              -------
SMITH                  400
```

Which SQL clause completes the program and generates the desired output?

A. where calculated Average > (select avg(Cost) from WORK.ONE)

B. having Average   > (select avg(Cost) from WORK.ONE)

C. having avg(Cost) < (select avg(Cost) from WORK.ONE)

D. where avg(Cost) > (select avg(Cost) from WORK.ONE)

Explanation:

C: should be >;

A/D: ERROR: Summary functions are restricted to the SELECT and HAVING clauses only.

44. Which dictionary table provides information on each occurrence of the variable named LastName?

A. DICTIONARY.TABLES

B. DICTIONARY.COLUMNS

C. DICTIONARY.MEMBERS

D. DICTIONARY.VARIABLES

Explanation:

Dictionary tables are commonly used to monitor and manage SAS sessions because the data is easier to manipulate than the output from procedures such as PROC DATASETS.

Dictionary tables are special, read-only SAS tables that contain information about SAS data libraries, SAS macros, and external files that are in use or available in the current SAS session. Dictionary tables also contain the settings for SAS system options and SAS titles and footnotes that are currently in effect. For example, the DICTIONARY.COLUMNS table contains information (such as name, type, length, and format) about all columns in all tables that are known to the current SAS session.

Dictionary tables are

- created each time they are referenced in a SAS program
- updated automatically
- limited to read-only access.

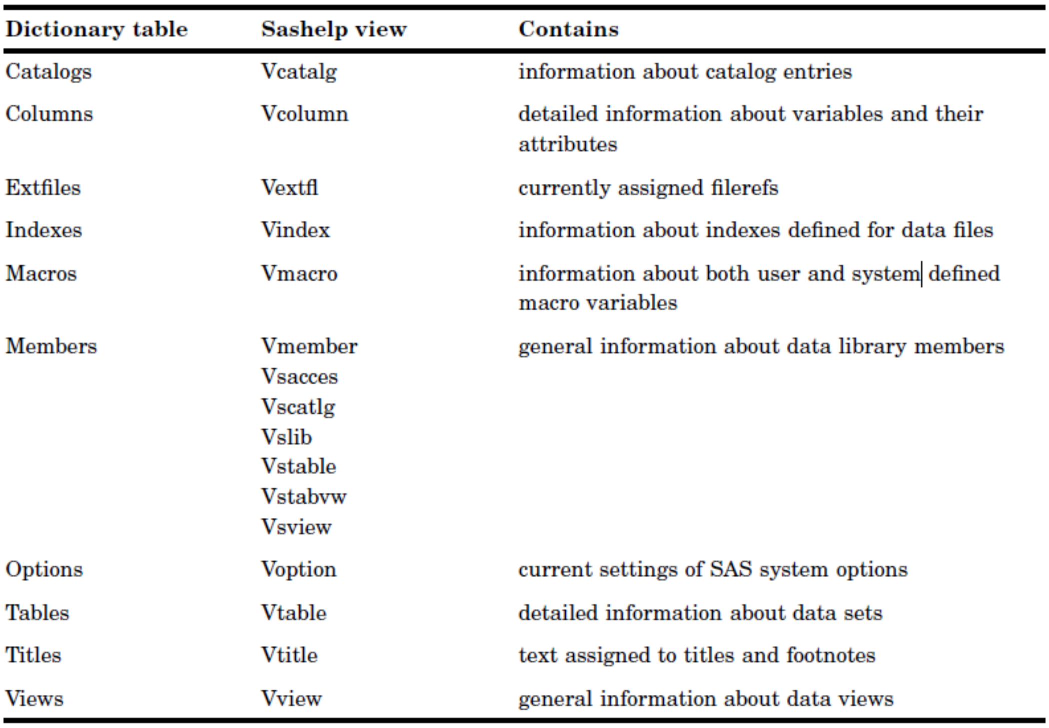
A. DICTIONARY.TABLES: detailed information about data sets;

B. DICTIONARY.COLUMNS: detailed information about variables and their attributes;

C. DICTIONARY.MEMBERS: general information about data library member;

D. DICTIONARY.VARIABLES: not valid.

The following table describes some of the Dictionary tables that are available and lists the corresponding SASHELP views.



Dictionary table	Sashelp view	Contains
Catalogs	Vcatalg	information about catalog entries
Columns	Vcolumn	detailed information about variables and their attributes
Extfiles	Vextfl	currently assigned filerefs
Indexes	Vindex	information about indexes defined for data files
Macros	Vmacro	information about both user and system defined macro variables
Members	Vmember Vsacces Vscatlg Vslib Vstable Vstabvw Vsview	general information about data library members
Options	Voption	current settings of SAS system options
Tables	Vtable	detailed information about data sets
Titles	Vtitle	text assigned to titles and footnotes
Views	Vview	general information about data views

45. To create a list of unique Customer_Id values from the customer data set, which of the following techniques can be used?

technique 1: proc SORT with NODUPKEY and OUT=
technique 2: data step with IF FIRST.Customer_Id=1
technique 3: proc SQL with the SELECT DISTINCT statement

A. only technique 1

B. techniques 1 and 2

C. techniques 1 and 3

D. techniques 1, 2, or 3

Explanation:

technique 1: The NODUPKEY option checks for and eliminates observations that have duplicate BY-variable values.

technique 2: 需要实现对 FIRST 变量 SORT，当然，如果我们假设条件里面隐含得给出了 presort

的事实，那么，第二个方法是可行的。见 52 题.
technique 3: select unique observations.

46. Given the SAS data sets:

```
WORK.CLASS1                  WORK.CLASS2

Name        Course             Name        Class
------      ------           ------      -----
Lauren      MATH1              Smith       MATH2
Patel       MATH1              Farmer      MATH2
Chang       MATH1              Patel       MATH2
                               Hillier     MATH2
```

The following SAS program is submitted:

```
proc sql;
    select Name
    from WORK.CLASS1
    [_insert_set_operator_]
    select Name
    from WORK.CLASS2
    ;
quit;
```

The following output is desired:

```
              Name
              ------
              Chang
              Lauren
```

Which SQL set operator completes the program and generates the desired output?

A. intersect corr
B. except
C. intersect
D. left except

47. The following SAS program is submitted:

```
 %macro execute;
    [_insert_statement_here_]
        proc print data=SASUSER.HOUSES;
        run;
    %end;
%mend execute;
%execute
```

Which statement completes the program so that the PROC PRINT step executes on Thursday?

A. if &sysday = Thursday then %do;

B. %if &sysday = Thursday %then %do;

C. %if "&sysday" = Thursday %then %do;

D. %if &sysday = "Thursday" %then %do;

Explanation

Macro variables are only saved as character string. When determine the value of macro variable, quotation marks for character string is not needed.

Also, pay attention to % sign.

48. Given the following program and data:

```
data WORK.BDAYINFO;
   infile datalines;
   input Name $ Birthday : mmddyy10.;
datalines;
Alan    11/15/1950
Barb    08/23/1966
Carl    09/01/1963
;
run;

%let Want=23AUG1966;
proc print data=WORK.BDAYINFO;
   [_insert_statement_]
run;
```

What is the WHERE statement that successfully completes the PROC PRINT and selects the observation for Barb?

A. where Birthday=&Want;

B. where Birthday="&Want";

C. where Birthday="&Want"d;

D. where Birthday='&Want'd;

49. Which macro statement would remove the macro variable Mv_Info from the symbol table?

A. %mdelete &Mv_Info;

B. %symerase Mv_Info;

C. %symdel &Mv_Info;

D. %symdel Mv_Info;

50. The table WORK.PILOTS contains the following data:

```
    Id     Name        Jobcode    Salary
   ---    ------     -------    ------
   001    Albert     PT1           50000
   002    Brenda      PT1           70000
   003    Carl        PT1           60000
   004    Donna        PT2           80000
   005    Edward      PT2           90000
   006    Flora       PT3          100000
```

A query was constructed to display the pilot salary means at each level of Jobcode and the difference to the overall mean salary:

```
Jobcode       Average    Difference
---------------------------------------------
PT1                60000          -15000
PT2                85000           10000
PT3               100000           25000
```

Which select statement could NOT have produced this output?

A. select

```
   Jobcode,
   avg(Salary) as Average,
   calculated Average - Overall as difference
from
   WORK.PILOTS,
   (select avg(Salary) as Overall from WORK.PILOTS)
group by jobcode
;
```

B. select

```
   Jobcode,
   avg(Salary) as Average,
   (select avg(Salary) from WORK.PILOTS) as Overall,
   calculated Average - Overall as Difference
from WORK.PILOTS
group by 1
;
```

C. select

```
   Jobcode,
     Average,
     Average-Overall as Difference
from
     (select Jobcode, avg(Salary) as Average
     from WORK.PILOTS
     group by 1),
     (select avg(Salary) as Overall
     from WORK.PILOTS)
;
D. select
     Jobcode,
     avg(Salary) as Average,
     calculated Average-(select avg(Salary) from WORK.PILOTS)
          as Difference
from WORK.PILOTS
group by 1
;
```

Explanation:

In one select statement, 前面刚刚定义的 variable，下一句要在运算中用，就要加 prefix：calculated，否则系统找不到，因为在 compile 阶段会出错。If the FROM clause includes a dataset which have defined that generated variable, then in select clause it could be referenced directly without prefix ‘calculated’.

B: ERROR Message: The following columns were not found in the contributing tables: Overall. The fifth line should be corrected as below:

calculated Average - calculated Overall as Difference.

51. The SAS data set WORK.TEMP is indexed on the variable Id:

```
Id   Amount
--   ------
P       52
P       45
A       13
A       56
R       34
R       12
R       78
```

The following SAS program is submitted:

```
proc print data=WORK.TEMP;
     [_insert_BY_statement_]
run;
```

Which BY statement completes the program, creates a listing report that is grouped by Id, and completes without errors?

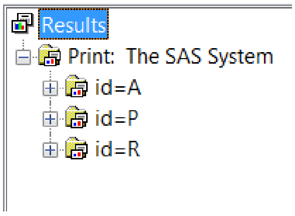
A. by Id;

B. by Id grouped;

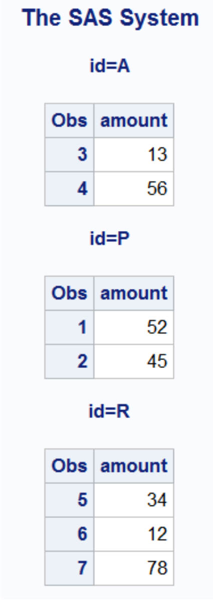
C. by Id descending;

D. by descending Id;

A:



Results
Print: The SAS System
id=A
id=P
id=R



The SAS System

id=A

Obs	amount
3	13
4	56

id=P

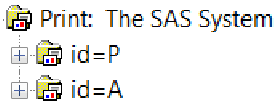
Obs	amount
1	52
2	45

id=R

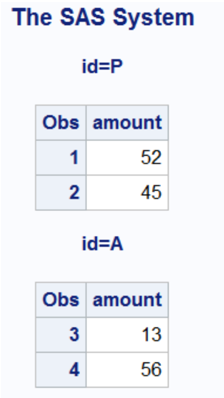
Obs	amount
5	34
6	12
7	78

B and C: ERROR: Variable GROUPED/ descending not found.

D:



Print: The SAS System
id=P
id=A



The SAS System

id=P

Obs	amount
1	52
2	45

id=A

Obs	amount
3	13
4	56

2012/11/17 日考试中，这道题有所改动，是直接给了 by id 的结果，问你 proc print 的运行情况。

52. To create a dataset with unique values of a given variable using a data step and the FIRST. VARIABLES and LAST.VARIABLES, it is assumed that the input dataset is:

A. sorted on that variable.
B. indexed by that variable.
C. naturally in order.
D. any of the above A, B, or C

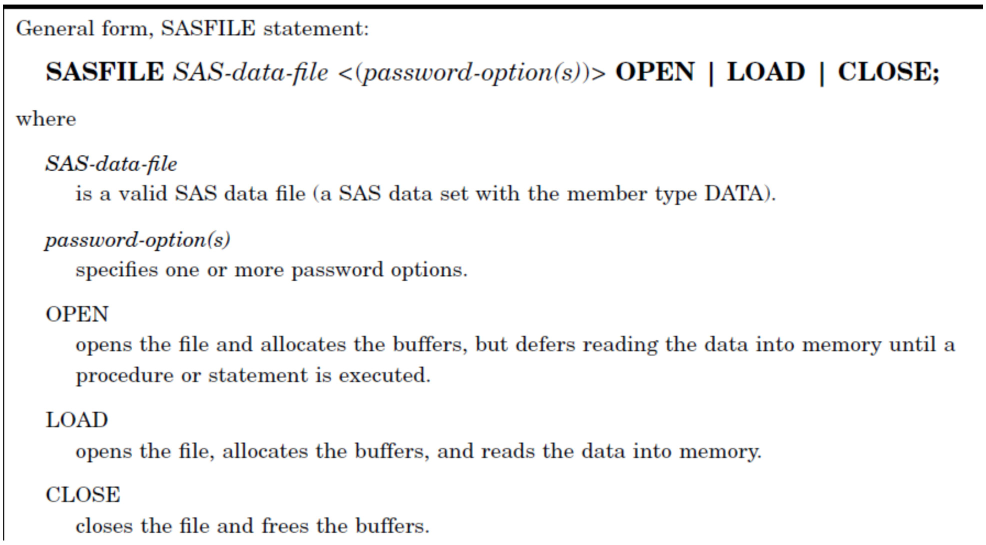
53.The SASFILE statement requests that a SAS data set be opened and loaded into memory:

A. one page at a time.
B. one variable at a time.
C. one observation at a time.
D. in its entirety, if possible.

Explanation:

The SASFILE statement opens a SAS data file and allocates enough buffers to hold the entire file in memory. Once the data file is read, the data is held in memory, and it is available to subsequent DATA and PROC steps or applications until either

- a SASFILE CLOSE statement frees the buffers and closes the file
- the program ends, which automatically frees the buffers and closes the file.



General form, SASFILE statement:

SASFILE *SAS-data-file* <(*password-option(s)*)> **OPEN | LOAD | CLOSE;**

where

SAS-data-file
is a valid SAS data file (a SAS data set with the member type DATA).

password-option(s)
specifies one or more password options.

OPEN
opens the file and allocates the buffers, but defers reading the data into memory until a procedure or statement is executed.

LOAD
opens the file, allocates the buffers, and reads the data into memory.

CLOSE
closes the file and frees the buffers.

54. The following SAS program is submitted:

```
   %let Name1=Shoes;
  %let Name2=Clothes;
  %let Root=name;
  %let Suffix=2;
  %put &&&Root&Suffix;
```

What is written to the SAS log?

A. &Name2

B. Clothes
C. &&&Root&Suffix
D. WARNING: Apparent symbolic reference ROOT2 not resolved.

55. Given the SAS data sets:

```
WORK.ONE                        WORK.TWO

Year   Qtr   Budget          Year   Qtr   Sales
----   ---   ------          ----   ---   -----
2001    3      500            2001    4     300
2001    4      400            2002    1     600
2003    1      350
```

The following SAS program is submitted:

```
  proc sql;
     select
          TWO.*,
          budget
     from
          WORK.ONE
          [_insert_join_operator_]
          WORK.TWO
     on ONE.Year=TWO.Year
     ;
quit;
```

The following output is desired:

```
Year   Qtr   Sales     Budget
----   ---   -----     ------
2001    4     300         500
2001    4     300         400
2002    1     600           .
   .     .       .        350
```

Which join operator completes the program and generates the desired output?

A. left join
B. right join
C. full join
D. outer join

Explanation:

D: not a valid clause.

56. The SAS data set WORK.ADDRESSES contains the email addresses of The XYZ Corporation's customers in a variable named Email_Address. The following DATA step is submitted:

```
    data   _null_;
      set WORK.ADDRESSES;
      [_insert_statement_]
      put "filename mail email '" Email_Address   "'; ";
      put "data _null_;";
      put "      file mail;";
      put "      put 'Thank you for your continued';";
      put "      put 'support of The XYZ Corporation.';";
      put "      put 'We appreciate your patronage.';";
      put "      put 'Sincerely,';" ;
      put "      put 'The XYZ Corporation';";
      put "run;" ;
   run;
```

Which statement completes the program and **creates a SAS program file?**

A. infile "c:\email.sas";
B. output "c:\email.sas";
C. file   "c:\email.sas";
D. None of the above.

57. Which of the following is true about the COMPRESS=YES data set option?

A. It uses the Ross Data Compression method to compress numeric data.
B. It is most effective with character data that contains repeated characters.
C. It is most effective with numeric data that represents large numeric values.
D. It is most effective with character data that contains patterns, rather than simple repetitions.

Explanation:

A: it uses run-length encoding,游程编码

58. Given the SAS dataset WORK.ONE:

```
   Salary
  ------
     200
     205
       .
     523
```

The following SAS program is submitted:

```
proc sql;
      select *
      from WORK.ONE
      [_insert_where_clause_]
      ;
quit;
```

The following output is desired:

```
Salary
------
      200
      205
      523
```

Which WHERE expression completes the program and generates the desired output?

A. where Salary is not .

B. where Salary ne missing

C. where Salary ne null

D. where Salary is not missing

Explanation:

To correct A: "where salary ne .".

59. The SAS data set WORK.TEST has an index on the variable Id and the following SAS program is submitted.

```
data WORK.TEST;
      set WORK.TEST( keep=Id Var_1 Var_2 rename=(Id=Id_Code));
      Total=sum(Var_1, Var_2);
run;
```

Which describes the result of submitting the SAS program?

A. The index on Id is deleted.

B. The index on Id is updated as an index on Id_Code.

C. The index on Id is deleted and an index on Id_Code is created.

D. The index on Id is recreated as an index on Id_Code.

Explanation:

HOW TO KEEP INDEX WHEN RENAME THE VARIABLE USED AS INDEX

```
PROC DATASETS LIB=WORK NOLIST;
MODIFY TEMP;
RENAME ID=ID_CODE;
QUIT;
```

60. Given the data set SASHELP.CLASS:

```
Name          Age
-------      ---
Mary           15
Philip        16
Robert        12
Ronald        15
```

The following SAS program is submitted:

```
%macro MP_ONE(pname=means);
    proc &pname data=SASHELP.CLASS;
    run;
%mend;
%MP_ONE(print)
%MP_ONE()
```

Which PROC steps execute successfully?

A. PROC MEANS only
B. PROC PRINT only
C. PROC MEANS and PROC PRINT
D. No PROC steps execute successfully

61. In a data step merge, the BY variables in all data sets must have the same:

A. name.
B. name and type.
C. name and length.
D. name, type, and length.

Explanation:

The BY-variables must have the same name and type, but the lengths do not have to be the same, the variable in merged data will use the length in the first listed data set in MERGE statement.

62. Given the following macro program and invocation:

```
%macro MAKEPGM(NEWNAME, SETNAME);
```

```
      data &NEWNAME;
         set &SETNAME;
      run;
      %put ---> inside macro &NEWNAME &SETNAME;
   %mend;

   %MAKEPGM(WORK.NEW, SASHELP.CLASS)
   %put ---> outside macro &NEWNAME &SETNAME;
```

Which of these choices shows the correct %PUT statement output if the program is submitted at the beginning of a new SAS session? Note that other lines may be written to the SAS log by the program, but only the %PUT output is shown here.

A. ---> inside macro WORK.NEW SASHELP.CLASS
   ---> outside invocation WORK.NEW SASHELP.CLASS
B. ---> inside macro WORK.NEW SASHELP.CLASS
   ---> outside invocation &NEWNAME &SETNAME
C. ---> inside macro &NEWNAME &SETNAME
   ---> outside invocation WORK.NEW SASHELP.CLASS
D. ---> inside macro &NEWNAME &SETNAME
   ---> outside invocation &NEWNAME &SETNAME

63. The following SAS program is submitted:

```
   %macro COLS1;
      Name Age;
   %mend;
   %macro COLS2;
      Height Weight;
   %mend;
   proc print data=SASHELP.CLASS;
      [_insert_VAR_statement_here_]
   run;
```

Which VAR statement successfully completes the program to produce a report containing four variables?

A. var %COLS1 %COLS2;
B. var %COLS1-%COLS2;
C. var %COLS1 Weight Height;
D. var Weight Height %COLS1;

Error message:
A:

```
488      proc print data=SASHELP.CLASS;
MLOGIC(COLS1):   Beginning execution.
489      var %COLS1 %COLS2;
MLOGIC(COLS1):   Ending execution.
MLOGIC(COLS2):   Beginning execution.
NOTE: Line generated by the invoked macro "COLS2".
1        Height Weight; ONLY second macro variable is resolved.
         ------
         180
MLOGIC(COLS2): Ending execution.
ERROR 180-322: Statement is not valid or it is used out of proper order.
490      run;
```

B:

```
493      proc print data=SASHELP.CLASS;
MLOGIC(COLS1):   Beginning execution.
494      var
494! %COLS1-%COLS2;
MLOGIC(COLS1):   Ending execution.
MLOGIC(COLS2):   Beginning execution.
494                   -%COLS2; symbol '-' cannot be used in the list of macro variables.
                       ------
                      180
MLOGIC(COLS2):   Ending execution.
ERROR 180-322: Statement is not valid or it is used out of proper order.
```

C:

```
498      proc print data=SASHELP.CLASS;
MLOGIC(COLS1):   Beginning execution.
499      var %COLS1 Weight Height;
                       ------
                       180
MLOGIC(COLS1):   Ending execution.
ERROR 180-322: Statement is not valid or it is used out of proper order.
500      run;
```

If there are macro variables and regular variable name, regular variable names should be listed first, then list macro variables.

If there are regular variables only or one macro variables only, then no error will show.

Answer

1.A

2.A

3.C

4.C

5.D

6.C

7.A 【C→sort size】

8.B

9.D

10.D

11.C

12.D

13.C

14.D

15.A

16.C

17.D

18.A

19.D

20.A

21.A

22.C

23.B

24.B

25.C

26.A

27.A

28.A

29.B

30.C

31.D

32.D

33.B

34.C

35.B

36.B

37.B

38.B

39.C

40.B

41.C

42.B

43.B

44.B

45.C

46.B

47.B

48.C

49.D

50.B

51.A

52.A

53.D

54.B

55.C

56.C

57.B

58.D

59.A

60.A

61.B

62.B

63.D