

**TASS Feb 9, 2018**

**Who wants to be a SAS Programmer**

**Questions and Answers**



# Rules of the Game

- ◆ 7      **Advanced SAS**      **(top prize)**
- 6      **Advanced SAS**
- 5      **Advanced SAS**
- ◆ 4      **SAS Base**      **(2<sup>st</sup> prize)**
- 3      **SAS Base**
- ◆ 2      **Friends of SAS**      **(1<sup>st</sup> prize)**
- 1      **Friends of SAS**



In this partial PROC CONTENTS output, what is the default length of the variable *Month*?

Alphabetic List of Variables and Attributes			
#	Variable	Type	Length
1	Month	Num	???



A.

2 bytes

B.

8 bytes

C.

16 or 17 bytes

D.

32,767 bytes

In this partial PROC CONTENTS output, what is the default length of the variable *Month*?

Alphabetic List of Variables and Attributes			
#	Variable	Type	Length
1	Month	Num	???



A.

2 bytes

B.

8 bytes

C.

16 or 17 bytes

D.

32,767 bytes

## Solution to Question 1:

**Month** is a numeric variable, and the default length of numeric variables is 8 bytes.

Which LIBNAME statement has the correct syntax?



A. libname reports 's:\workshop';

B. libname orion s:\workshop;

C. libname 3456a 's:\workshop';

D. libname reports2018 's:\workshop';

Which LIBNAME statement has the correct syntax?



A. libname reports 's:\workshop';

B. libname orion s:\workshop;

C. libname 3456a 's:\workshop';

D. libname reports2018 's:\workshop';



## Solution to Question 2:

In a basic LIBNAME statement, you specify the keyword LIBNAME, a valid *libref* (maximum length of 8 characters), and then the physical name of the library in quotation marks.

**Which of the following displays the contents of an external file from within a SAS session?**



**A.** the LIST procedure

**B.** the PRINT procedure

**C.** the FSLIST procedure

**D.** the VIEWTABLE window

Which of the following displays the contents of an external file from within a SAS session?



A. the LIST procedure

B. the PRINT procedure

C. the FSLIST procedure

D. the VIEWTABLE window

### Solution to Question 3:

The PRINT procedure and VIEWTABLE window display the values in SAS data sets. The FSLIST procedure displays the values in external files. There is no LIST procedure in SAS.

The following SAS program is submitted:

```
data names;  
  title='EDU';  
  if title='EDU' then Division='Education';  
  else if title='HR' then Division='Human Resources';  
  else Division='Unknown';  
run;
```

Which of the following represents the value of the variable *Division* on the output data set?



A.

*Educatio*

B.

*Education*

C.

*Human Re*

D.

*Human Resources*

The following SAS program is submitted:

```
data names;  
  title='EDU';  
  if title='EDU' then Division='Education';  
  else if title='HR' then Division='Human Resources';  
  else Division='Unknown';  
run;
```

Which of the following represents the value of the variable *Division* on the output data set?



A.

*Educatio*

B.

*Education*

C.

*Human Re*

D.

*Human Resources*

## Solution to Question 4:

The length of the variable Division is set to 9 when the DATA step compiles. Since the value of the variable Title is *EDU*, the first IF condition is true; therefore, the value of the variable Division is *Education*.

At the beginning of a new SAS session, the following program is submitted:

```
%macro one;  
  %let proc= means;  
  proc &proc data=sashelp.class;  
%mend;  
%one
```

Where is the macro variable *proc* stored?



A. the local symbol table

B. the global symbol table

C. in the SAS data set  
**Work.Sasmacr**

D. the program fails to execute  
because PROC is a reserved word



At the beginning of a new SAS session, the following program is submitted:

```
%macro one;  
  %let proc= means;  
  proc &proc data=sashelp.class;  
%mend;  
%one
```

Where is the macro variable *proc* stored?



A. the local symbol table

B. the global symbol table

C. in the SAS data set  
**Work.Sasmacr**

D. the program fails to execute  
because PROC is a reserved word

## Solution to Question 5:

New macro variables that are created during macro execution are created in the macro's local symbol table. Macro variables that are created with a %LET statement in open code are stored in the global symbol table. PROC is not a reserved word in the SAS macro facility.

**The following SAS program is submitted:**

```
%let var= chicago, 1;  
data one;  
  var= 'new york, 2';  
  newvar= %scan(&var, 2, %bquote(,));  
run;
```

**Why does the program fail to execute?**



**A.** %bquote(,) is invalid syntax

**B.** the %SCAN function does not exist

**C.** the %SCAN function has too many arguments

**D.** the macro variable *var* does not get created properly

**The following SAS program is submitted:**

```
%let var= chicago, 1;  
data one;  
  var= 'new york, 2';  
  newvar= %scan(&var, 2, %bquote(,));  
run;
```

**Why does the program fail to execute?**



**A.** %bquote(,) is invalid syntax

**B.** the %SCAN function does not exist

**C.** the %SCAN function has too many arguments

**D.** the macro variable *var* does not get created properly

## Solution to Question 6:

First, the macro processor resolves the macro variable reference `&var` to a value of *chicago, 1*. Then, the `%SCAN` function is interpreted as `%scan(chicago, 1, 2, %bquote(,))`. As a result, the otherwise valid syntax `%bquote(,)` appears to be a fourth argument to the function, and `%SCAN` accepts a maximum of three arguments.

The format \$mine. was created and stored in the F.X catalog. Given the following SAS code:

```
libname A 'SAS library location';  
libname F 'SAS library location';  
options fmtsearch=(A, F.X);
```

What is the first location searched for the \$mine. format?



A.

F.X

B.

A.Formats

C.

Work.Formats

D.

Library.Formats

The format \$mine. was created and stored in the F.X catalog. Given the following SAS code:

```
libname A 'SAS library location';  
libname F 'SAS library location';  
options fmtsearch=(A, F.X);
```

What is the first location searched for the \$mine. format?



A.

F.X

B.

A.Formats

C.

Work.Formats

D.

Library.Formats

## Solution to Question 7:

SAS searches for formats first in the Work.Formats catalog, unless it appears in the FMTSEARCH list. Then the FMTSEARCH locations are searched in the order they appear. According to the FMTSEARCH= option in this example, the search path sequence for the user-defined formats is: Work.Formats, Library.Formats, A.Formats, F.X. If the format is not located, an error message is written to the SAS log.



**Thank you playing  
Who Wants to be a SAS Programmer**





# Rules of the Game

- ◆ 7      **Advanced SAS**      **(top prize)**
- 6      **Advanced SAS**
- 5      **Advanced SAS**
- ◆ 4      **SAS Base**      **(2<sup>st</sup> prize)**
- 3      **SAS Base**
- ◆ 2      **Friends of SAS**      **(1<sup>st</sup> prize)**
- 1      **Friends of SAS**



**What result would you expect from submitting this step?**

```
proc print data=work.newsalesemps  
run;
```



**A.** an HTML report of the work.newsalesemps data set

**B.** an error message in the log

**C.** a LISTING report of the work.newsalesemps data set

**D.** the creation of a temporary data set called work.newsalesemps

**What result would you expect from submitting this step?**

```
proc print data=work.newsalesemps  
run;
```



**A.** an HTML report of the work.newsalesemps data set

**B.** an error message in the log

**C.** a LISTING report of the work.newsalesemps data set

**D.** the creation of a temporary data set called work.newsalesemps

## Solution to Question 1:

There is a missing semicolon following the data set name. When this step runs, SAS will interpret the word RUN as an option in the PROC PRINT statement (because of the missing semicolon). As a result, the PROC PRINT step will not execute and an error message will be displayed in the log.

Suppose you submit a short, simple DATA step. If the active window displays the message **DATA step running** for a long time, what probably happened?



A. you misspelled a keyword

B. you forgot to end the DATA step with a RUN statement

C. you specified an invalid data set option

D. some data values were not appropriate for the SAS statement that you specified

Suppose you submit a short, simple DATA step. If the active window displays the message **DATA step running** for a long time, what probably happened?



A. you misspelled a keyword

B. you forgot to end the DATA step with a RUN statement

C. you specified an invalid data set option

D. some data values were not appropriate for the SAS statement that you specified



## Solution to Question 2:

Without a RUN statement (or a following DATA or PROC step), the DATA step doesn't execute. Unbalanced quotation marks can also cause the **DATA step running** message if relatively little code follows the unbalanced quotation mark. The other three problems above generate errors in the Log window.

The following SAS program is submitted:

```
data newstaff;  
  set staff;  
  <insert WHERE statement here>  
run;
```

Which of the following WHERE statements completes the program and selects only observations with a *hire\_date* of *February 23, 2000*?



A. where hire\_date='23feb2000'd;

B. where hire\_date='23feb2000';

C. where hire\_date='02/23/2000'd;

D. where hire\_date='02/23/2000';

The following SAS program is submitted:

```
data newstaff;  
  set staff;  
  <insert WHERE statement here>  
run;
```

Which of the following WHERE statements completes the program and selects only observations with a *hire\_date* of *February 23, 2000*?



A. where hire\_date='23feb2000'd;

B. where hire\_date='23feb2000';

C. where hire\_date='02/23/2000'd;

D. where hire\_date='02/23/2000';

## Solution to Question 3:

A SAS date constant must take the form of one- or two-digit day, three-digit month, and two- or four-digit year, enclosed in quotation marks and followed by a **d** (*'ddmmyy<yy>d'*).

**Which of the following SAS date formats displays the SAS date value for January 16, 2002 in the form of 16/01/2002?**



**A.** DATE10.

**B.** DDMMYY10.

**C.** WEEKDATE10.

**D.** DDMMYYYY10.

**Which of the following SAS date formats displays the SAS date value for January 16, 2002 in the form of 16/01/2002?**



**A.**

DATE10.

**B.**

DDMMYY10.

**C.**

WEEKDATE10.

**D.**

DDMMYYYY10.

## Solution to Question 4:

The requested output is in day-month-year order and is 10 bytes long, so `DDMMYY10.` is the correct format. Although `WEEKDATE10.` is a valid SAS format, it does not display the SAS date value as shown in the question above. `DDMMYYYY10.` is not a valid SAS date format, and the `DATEw.` format cannot accept a length of 10.

The following program is submitted:

```
proc sql noprint;  
  select name  
    into: info separated by ' '  
    where libname="SASHELP" and  
           memname="CLASS";  
quit;
```

Given that the SAS data set **SASHELP.CLASS** exists, which of the following items is generated by this program?



**A.** a list of names

**B.** a syntax error in the log

**C.** a macro variable named *info*

**D.** a report that shows metadata information



The following program is submitted:

```
proc sql noprint;  
  select name  
    into: info separated by ' '  
    where libname="SASHELP" and  
           memname="CLASS";  
quit;
```

Given that the SAS data set **SASHELP.CLASS** exists, which of the following items is generated by this program?



A. a list of names

B. a syntax error in the log

C. a macro variable named *info*

D. a report that shows metadata information

## Solution to Question 5:

The INTO clause is used to create macro variables in a PROC SQL step. In this program, a list of column names, separated by spaces, is written as the value of the macro variable info. No report is generated because of the NOPRINT option in the PROC SQL statement.

The following program is submitted:

```
data two;  
  y='2';  
run;  
  
%let x=10;  
%let var=y;  
  
data one;  
  set two (keep=&var);  
  z=&var*&x;  
run;
```

Which of the following answers is the value of variable `z` when the program finishes executing?



A. `_ERROR_`

B. `20`

C. `y*10`

D. `.` (missing numeric value)

The following program is submitted:

```
data two;  
  y='2';  
run;  
  
%let x=10;  
%let var=y;  
  
data one;  
  set two (keep=&var);  
  z=&var*&x;  
run;
```

Which of the following answers is the value of variable `z` when the program finishes executing?



A. `_ERROR_`

B. `20`

C. `y*10`

D. `.` (missing numeric value)

## Solution to Question 6:

The macro variable reference `&var` resolves to `Y` in the `SET` statement. Then, the macro variable reference `&var` resolves to `Y` in the assignment statement, and `Y` is a character variable with the value of `2`. The macro variable reference `&x` resolves to a numeric value of `10`. SAS performs automatic character-to-numeric conversion on the value of the variable `Y`. Therefore, the value of the variable `Z` is `20`.

The following program is submitted:

```
%macro loop;  
  %let i=1;  
  %do %until (&i ge 3);  
    %put I is &i;  
    %let i=%eval(&i +1);  
  %end;  
%mend;  
%loop
```

How many times does the %PUT statement write to the SAS log?



A.

0

B.

1

C.

2

D.

3

The following program is submitted:

```
%macro loop;  
  %let i=1;  
  %do %until (&i ge 3);  
    %put I is &i;  
    %let i=%eval(&i +1);  
  %end;  
%mend;  
%loop
```

How many times does the %PUT statement write to the SAS log?



A.

0

B.

1

C.

2

D.

3

## Solution to Question 7:

During the first two iterations of the loop, the %PUT statement writes `i is 1`, and `i is 2` to the SAS log. After the second iteration, `i` increments from 2 to 3, satisfying the expression in the %DO %UNTIL statement, so the loop does not iterate again. Therefore, the %PUT statement writes two lines to the SAS log.



**Thank you playing  
Who Wants to be a SAS Programmer**

