

1.B

1)temp is declared as a permanent lib so permanent dataset will be created

2)same dataset can be read and write in a step

3)work and perm datasets can be merged

2.B

OBS option will let SAS know last observation to process. FIRSTOBS option will let SAS know first observation to process. So for first proc print FIRSTOBS =100 and OBS (i like to call it LASTOBS) = 500, so there are 401 observation (including 100th observation) processed by it.

Now for proc means FIRSTOBS=500 and OBS (or LASTOBS) = Max (or 5000 in this case), so there are 4501 observation (including 500th observation) is processed by it.

3.A

>>always "Dog" will be assigned to word as amount is hardcoded 7

in the beginning of each

iteration

>>and a value of 5 is saved in amount as it is hardcoded before the end of each output.

5.C

do UNTIL loop will execute once no matter what is the result of prod gt 6 condition. Also do until will check for condition at end of the loop. Now with this knowledge read the above answer.

7.C

If however, \$25.31 did begin at column 10 then there would be a missing value in dataset inventory, this is due to invalid data for price (\$ is a character and is not accepted into a numeric variable without some form of informat e.g DOLLAR.)

8.B

reason:by default sas proceeds to next column. here after reading 5 columns for type the position

of the pointer is at y. due to +1 cursor moves to one column more

9. `coz.dlm=','` is a infile option.

10.B

as we are mentioning the fileref and not external file.

becoz of the filename statement mentioned in the program, there is no need to enclose `rawdata2` in quotation marks.

11.D

`_end` is sas keyword which will become true when SAS reads last record of a dataset. This value you cannot use directly in your program, so we create a alias name `eof` (end of file), but you can name it anything. `EOF` will carry the same value as internal variable `END`. So as we know `1=true` and `0=false`. if `EOF = 1;` will output only the last observation

12.D

with `DSD` option two consecutive delimiters are not consider as one, so missing value is assigned to `item2` and as `item2` is character variable it has missing character value `" "`..without `dsd` it will be error.

13.D.

Reason being statement `"input style $ @;"` will hold input stream until next input statement. Program will read next input statement only if `style` is `condo` or `ranch`. But even when `style` is not `condo` or `ranch` when run statement is executed output will be written with just `style` variable having value and all others will be missing.

14.B

because once you tell sas some condition as we have here if `style = 'CONDO'` or `style = 'RANCH'`; then it want do anything else.

15.A

As each input statement other than last (input age) ends with @, SAS will not advance to next raw data record. Only after age is read, SAS will finish the dataset iteration and read next record.

16. Because in it a set statement not merge, so only inemp or insal can be true at a time.

17.B

Bruce 30 40000

Bruce 30 35000

Dan 35 37000

Dan 35 .

18.B

The Variables are: Fname, age, salary and totals (i.e 4 variables)

Note: Obs will not be counted here

19.D

Because the two data sets are merged by fname variable. You need to rename the name variable of the salary data set to fname. Correct way to rename is (rename = (original_var_name=new_var_name))

20.C

You need to have the DESCENDING option in the BY statements in both the PROC SORT steps and the DATA step. If you omit the DESCENDING option in the DATA step in the BY statement, you generate error messages about improperly sorted BY Variables.

21. If libname name is not specified, then the data will be stored in WORK.

22.

Reading from permanent dataset Sasuser.houses & writing to a permanent dataset temp.report

25. The numeric value of Jan 01, 1960 is 0.

27.A

For SAS certification we need to assume that YEARCUTOFF option is set to 1JAN1960. This date is represented as 0 in SAS date format. Now MDY function will change normal date to SAS date, we are passing 15JAN1960 as the parameter to mdy which is 14th day from 1JAN1960, hence answer is option A.

28.

The qtr function returns a value of 1-4 depending upon which quarter of the year the sales_date lies. Of the twelve months, three will be assigned qtr(sales_date) equal to 3, and another three equal to 4, giving a total of 6 observations in the output. (第三和第四季度共 6 个月)

29.

MMDDYY10. : MM/DD/YYYY

Sunday = 1 以此类推

30.

the only correct DATE INFORMATS are:

MMDDYY8. reads dates written as mm/dd/yy.

MMDDYY10. reads dates written as mm/dd/yyyy.

DATE7. reads dates in the form ddMMMyy.

DATE9. reads dates in the form ddMMMyyyy.

31.

DSD = ',' is invalid because by default DSD is a comma. If you use DSD alone it would work.

32.

Because @5 age 2. will not create space for itself but rather it will overwrite name \$15, which is written first. So Alfred written by name variable will become Alfr14

33.

infile and input statements are used to read data from raw data file. File and put statements are used to write data to raw data file. In our example later is happening, so a raw data file is created and as data step is creating a dataset (it is not data _null_), it will create an output dataset too. So the answer is C raw data file and group dataset.

34.

because file statement is missing. no file is specified to which the values can be written. so put statement writes it to the log.

35.

because no delimiter is specified at the end of the file stmt. so the default delimiter space is used. if you want comma as a delimiter, use the stmt : file 'file specification' dlm=','

36.

Obs – firstobs +1

37.

Following three observation satisfy condition if price ≤ 5 .

B132S 2.34 HARDWARE 300 10

R18KY2 1.29 SOFTWARE 25 5

DGTY23 4.55 HARDWARE 67 2

Out of the above 3 observation only 2 observation have producttype = 'HARDWARE', so number of observation in HWARE is 2

38.

because the output statement is inside the do loop, an observation will be created each time through the loop, so 5 observations, 5 loops, 25 observations total

39.

Numeric missing values are the smallest.

Obs name age height

1 John . 35

2 June . 10

3 Tineke . 9

41.

VARNUM : Print a list of the variables by their logical position in the data set.

without varnum, proc contents will print the variables in alphabetic order. varnum makes proc contents to print the variables in the order they were created.

42.

PROC PRINT: data portion of data set

Proc FSLIST: enables you to browse external files that are not SAS data sets within a SAS session

PROC CONTENTS: descriptor portion of data set

43.

proc print only shows data portion of dataset and proc contents shows descriptor portion of a dataset.

44.

synatx:array array_name(len_ofarray) var_list;

var_list could be new or existing vars in datasets.

in the above program, dif_sales1- dif_sales3 are new.

bcos,jansales,febsales,marsales are existing in work.staff dataset.

diff_sales is the array name.

47.

What is a SAS array?

* a set of variables grouped together for the duration of a data step by being given a name in an ARRAY statement.

49.

it displays descending order for the first variable, so ans is d

50.

By default SAS will sort in ascending order, if you want descending then you specify descending keyword before the variable name. Now here we want ascending expense values within each ascending IDNUMBER value in our by statement we should have IDNUMBER first and then expenses.

52.

The word descending only modifies the variable directly after it.

so "by destination descending revenue" causes destination to be in ascending order

(A,B,C,D,E,F,G)and revenue will be in descending order (9,8,7,6,5,4,3).

55.

Notice that the LENGTH statement in the SELECT group has not been specified. Remember that without the LENGTH statement, values for Group might be truncated, as the first value for Group (London) is not the longest possible value.

See, city = 'London' - Length of assignment.

Second assignment will cut down the words when('CPH') city = 'Copenhagen'.

56. care about the capital CPH is in the when statement.

57. In A, this line in error: airplane type = 'Large' and staff = 10.

58. ** = exponentiation, X**Y raise X to the power of Y. So 3 at the power of 2 = 9

59.

reason: jobcategory='FA'; after this jobcategory variable's length is 2, it can hold only 2 bytes.

Eventhough there is no mistake in the step jobcategory=jobcategory || joblevel; jobcategory carries the value FA.

60. IF 语句在设置值的前面，无效

All summary function ignore missing value when computing values whereas operators like + - / * etc do not ignore missing. Thus SUM function will give non missing output as long as there is at least one variable with non missing value. Whereas the operator will always give missing as long as at least one variable in operation is missing.

61.

PDV initializes all variables in data step to missing first. So if statement yields true, so value of origpassenger becomes 100. Then transpassenger is assigned value of 100. Next step value of origpassenger is overwritten with missing value. So when you add using sum function which ignores missing value, values of origpassenger and transpassenger ie sum(.,100) gives you 100.

62. The variable jcode in the second dataset is renamed as JOBCODE

63. data(rename = (old-name-1=new-name-1 old-name-2=new-name-2 ... old-name-n=new-name-n))

64.

This is the log error after we run this program.

```
/* WARNING: Length of character variable has already been set. Use the LENGTH statement as  
the very first statement in the DATA STEP to declare the length of a character variable.*/
```

65.

The length of a variable always define automatically in the first statement and remain unchanged during the whole program until and unless we redefine it. So when DEPT1 is set the length will set to 5 and remain unchanged during the data step.

If set work.dept2 work.dept1; then the answer is 7.

66.

Length of a variable is assigned during first assignment, in this case description = "Senior Chemist" which is 14 character long, so the new variable description will have length of 14.

67.

when you use PAYROLL + WAGERATE, variable payroll will automatically be retained through-out the dataset. Its values is resetted at first observation of each department. The observations to output dataset total are output on last observation of each department. So number of observation in total dataset are number of departments in salary dataset. Moreover wagerate is getting added to payroll and that value is retained in payroll variable until the next observation of same department is read. After that second observations wagerate gets added to payroll, so on and so forth until all the 10 observations are read for a department. At last observation, payroll has the sum of all wagerate for that particular department and is outputted to total dataset.

68.

The most important statement in the program for this question is if last.department; We will get output only if last.department becomes true and we know there are five departments and data is sorted by department so last.department will become true 5 times and thus 5 observation in output.

69.

sum is a function which means it will need brackets () so A is not the correct answer.

There no such variable as grandtot, so program will give syntax error, so option B and D is not the correct answer.

C is the correct option because retain statement is creating a new variable and initializing it with zero. Sum function will add cost to grandtot for each observation and as grandtot is retained it will carry forward the added value.

70. Expression cannot ignore missing value, but the SUM function and statement can.

71.

The RETAIN statement

- is a compile-time only statement that creates variables if they do not already exist
- initializes the retained variable to missing before the first execution of the DATA step if you do not supply an initial value
- has no effect on variables that are read with SET, MERGE, or UPDATE statements.

72.

city has the same length with First, the length is 16, the letter of city is first 7 letter of first, the rest 9 digit is fix with blank (!! = one blank)

73.

By default the variable length is determined by the length of the first argument for **substr**.

So here we have $16 + 2(' ') + 7('England') = 25$ 原字符串 First = 'Ipswich, England' 长度为 16

注意:

在没有用 LENGTH 声明长度时, scan () 默认返回字符串的**存储长度为 200**。

trim() 去掉字符串尾部空格, 如果字符串为空, 则返回一个空格, 改变字符串的物理存储长度、字符长度

`trimn()` 去掉字符串尾部空格，如果字符串为空，则返回一个空字符串，改变字符串的物理存储长度、字符长度

`left()` 把字符串开头的空格移到尾部，**没有**改变字符串的物理存储长度,但字符长度被改变

`strip()` 去掉字符串开头和结尾的所有空格，改变字符串的物理存储长度、字符长度

`cat()` 与`||`作用类似，保留首尾全部空格

`catt()` 连接之前会去掉各字符串尾部空格，相当于连接符结合 `trim()`使用

`cats()` 连接之前会去掉首尾全部空格，相当于连接符结合 `strip()`或 `trim (left (字符串))` 使用

`catx()` 连接之前会去掉首尾全部空格，在字符串之间加上一个指定的字符串

74.

SCAN function: `Word = scan(title,3,' ');` Note that a blank and a comma are specified as delimiters.

You can also write the function without listing delimiters, because the blank and comma are default delimiters.

If there were NO space (' '), the value returned would be ' ' (missing character value). However, having the space before the delimiter (' ') returns the 'of', so it is 'B' instead of 'D'.

75.

Even if there is a space b/w the comma and the Agatha, it does not matter because 'blank' or space' is also one of the several(blank . < (+ | & ! \$ *) ^ - / , % default delimiters.

`scan(author,1,' ,')`就是在逗号后第一个单词

76.

A variable that is created by SUBSTR obtains its length from the length of argument. The default length of SCAN function is 200, which is the Length of argument. So answer is 200.

77.

D is incorrect as commas or OF is missing ...correct forms could be

`average = mean(num1, num2, num3 ,num4);` OR `average = mean(OF num1 num2 num3 num4);`

78.

Because the character value is automatically converted to numeric by SAS.

Here's the log results "NOTE: Character values have been converted to numeric values at the places given by: (Line):(Column)"

79.

for Product_Number Character expected but numerical got, so it automatically change it into Character by using best 12. beginning from right. So there are $12-4=8$ spaces before 5461

if you put a number which has 12 bytes then there is no space before that number. **1001/ 5461**

80. As numeric variable cannot be more than 8 bytes long Input=('13mar2000',date9.);

81.

ddmmyy10. formats date to be 10 bytes in length i.e dd/mm/yyyy, and 'put' converts value into a character type. Put=('13,mar2000'd, ddmmyy10.);

82.

The input statement reads the variable dept using 11 bytes. The value of dept (Department) consists of 8 bytes therefore 3 trailing blanks are present in the value of dept which need to be trimmed out using the trim function. This is concatenated to the put function which converts numeric variables to character variables using the supplied format (3.)

83.

input function converts character to numeric, put function converts numeric to char ...so here date is given as character,,input function and date is in the form of yymmdd10.

85.C

because of n+1 statement we assume a implicit retain n 0; statement.

First loop (0 lt 6) => true => n = 1

Second loop (1 lt 6) => true => n= 2

.....

last loop (6 lt 6) => false => do while exits and pdv writes 6 as value of n in output dataset.

86.D

A while condition isn't checked until the end of the loop, so the first time through, +1 is added to calls, resulting in 6- then the condition is checked and it stops.

87.

From SAS log:

NOTE: There were 3 observations read from the data set PRACTISE.BANKS.

NOTE: The data set PRACTISE.NEWBANK has 1 observations and 4 variables.

NOTE: DATA statement used (Total process time):

real time 0.07 seconds

cpu time 0.03 seconds

88.C LABEL name-1='姓名' name-2='标签 2' ... name-k='标签 k';

if you are using label statment inside proc print then you need to use label option on your proc print statement. This label inside a proc is applied temporarily and will replace any permanent labels.

89.D

It can not be C bcz the total width is 11 not 8, so it comes down to either of dollar11.2 or comma11.2. But comma11.2 does not insert dollar sign along with comma.

91.

这里的 define 后面的 '/', 其后面有很多选项来对变量格式, 输出方式等进行设置, 其选项如下:

GROUP: 分组显示

DISPLAY: 一般显示

ANALYSIS: 其值用于计算或统计

ORDER: 按其值排序显示

ACROSS: 交叉显示

COMPUTED: 其值在 compute 块里计算得到

93.C

if the style variable is present in id statement, you don't have to mention it in var statement, else you will get 2 style variable in the output, but since there is just 1 style column present ans is C

94.A

In the output where you can clearly see for variable val N is 3 not 4, so proc means only consider numeric non-missing values for that variable to calculate its statistics.

shoe_number 4 7.5000000

val 3 4.6666667

95. 为了呈现 2 各不同的表格: table var1; table var2;

97.A

Other commonly used options available in PROC MEANS include:

? DATA= Specify data set to use

? NOPRINT Do not print output

? MAXDEC=n Use n decimal places to print output

98.B

Titles and footnotes stay in effect until you replace them with new ones or cancel them with a null statement. The following null statement would cancel all current titles: TITLE;
When you specify a new title or footnote, it replaces the old title or footnote with the same number and cancels those with a higher number. For example, a new TITLE2 cancels an existing TITLE3, if there is one.

100. Option NONUMBER: Removes the page numbers from the output file.

102.B

Option DATE: Prints the date and TIME that the SAS session was initialized.

The option Time does not exist.

103.D

pagno=6 because proc print write 5 pages output.and after that proc means will write where..?obviously on page no =6;

Use the OPTIONS statement to change SAS system options.

The change(s) will remain in effect for the rest of the job/session or until changed again.

104. Bcause that value in between Fail(<50) and Pass(>51) Conditions

106. html close; Is the syntax to stop the output to be written to the ODS

107. if "CHEM3" = "chem3" is false because string comparision is case sensitive, hence answer is B.

108.B

because value of Jobcode="chem2" different of test value of Jobcode="Chem2" so THEN statement will be skiped and ELSE statment will be executed

109.C

because after reading following statement

```
input employee_name $ 1-4;
```

SAS advances to next line of raw data file and next input statement

```
if employee_name = 'Sue' then input age 7-8;
```

reads age for John as age of SUE which is 40.

110.B

Run the following:

```
data test;
```

```
input employee_name $ 1-4;
```

```
if employee_name = 'Ruth' then input idnum 10-11;
```

```
else input age 7-8;
```

```
put idnum= age=;
```

```
cards;
```

```
Ruth 39 11
```

```
Jose 32 22
```

```
Sue 30 33
```

```
John 40 44
```

```
;
```

```
run;
```

111.B

SAS is using 3 as a Boolean, and it's always TRUE. Because 3 is always equal to 3.

Basically you are writing and using the following code logic:

If $x=2$ or $3=3$ then expertise='Medium'; 4 is ALWAYS = 4, so this portion of the equation is always true.

So for a value other than missing, 1 and 2 the following happens regarding the logic. Take the value of 4 for instance:

If level=. (is false if level=4) go to the next ELSE IF.

If level=1 (is false if x=4) go to the next ELSE IF.

If level=2 or 3 then match. One OR the other must be true for this expression to be true (The value is 4, so level=2 is FALSE, but 3=3 is ALWAYS TRUE), so even though the value is 4, 3=3 is true and expertise='medium'. That's why you are getting the results of medium for the values of match when the values of level are anything but missing or 1.

To get the correct result that is medium for only 2 or 3, you should write the statement as below:

else if level=2 or level=3 then expertise='medium';

112.A

Since the 2. is not specified correctly, it considers 2 as column input and moves the pointer to column 2.

Also remember, if you just specified @4 weight; it would still be ok and it will give the correct value 95.

114. If we provide correct date constant date='01jan2000'd, will get the result.

115. As there is syntax error. "keep = product sales" which is the incorrect way to use KEEP option.

116.B

because retain statement for arrays should look something like retain monthsales1-monthsales12

117.B

Program will run without errors if we change keep = statement as

```
data work.totalsales(keep = msales1-msales12 )
```

```
set work.monthly (keep = sales);
```

```
array msales {12}
```

```
do i=1 to 12;
```

```
msales{i} = sales;  
end;  
run;
```

118.B

data errors occur in data step but it is a execution time error, so it continues to execute. Most of the syntax errors that occur during the compilation phase stop the execution of sas program

119. because the value is 1 or 0

120. Read in using Dollar format or read in as a character.

121. since a numeric value is expected for the variable been read, and a character value was found.

122.A

as the program has an error which is compilation error so this will flag `_ERROR_` to 1. i think the error occurs in "infile" statement.

right Input statment should be like below;

```
input @1 salary dollar6.;
```

or

```
input @1 salary comma6.;
```

123.D

Maximum Length of User-Supplied SAS Names SAS Language Element Maximum Length

Arrays 32

CALL routines 16

Catalog entries 32

DATA step statement labels 32

DATA step variable labels 256

DATA step variables 32

DATA step windows 32

Engines 8

Filerefs 8

Formats, character 31

Formats, numeric 32

Functions 16

Generation data sets 28

Informats, character 30

Informats, numeric 31

Librefs 8

Macro variables 32

Macro windows 32

Macros 32

Members of SAS data libraries (SAS data sets, views, catalogs, indexes) except for generation data sets 32

Passwords 8

Procedure names (first 8 characters must be unique, and may not begin with "SAS") 16

SCL variables 32

Variable name is not a problem,

1. Since division by zero, Mathematical operations could not be performed
2. The results of the operations have been set to missing values.