Improve Your SAS® Code Using User-written Functions

Bartosz Jabłoński

Warsaw University of Technology / Citi Europe PLC / ex-PAREXEL

yabwon@gmail.com

Friday, October 12th, 2018

PhUSE Single Day Event, Warsaw, Poland

https://www.linkedin.com/groups/8675451

ii2ts

(Informal) Introduction to The Subject

Art Carpenter's talk at SAS Global Forum 2018

Using Hash Tables to Manage Your Macro Language Control Files

Art Carpenter's talk at SAS Global Forum 2018

Using Hash Tables to Manage Your Macro Language Control Files

Video from SGF 2018 talk:
`https://video.sas.com/sasgf18/detail/videos/breakout-sessions/video/5766283050001/using-hash-tables-to-manage-your-macro-language-control-files?autoStart=false`

Final version of the paper:
`https://www.lexjansen.com/wuss/2018/41_Final_Paper_PDF.pdf`

The subject

How one can use `Proc FCMP`, User-written functions, and `%MACRO`s to use in-memory resident `Hash Tables` to share data between `datasteps`?

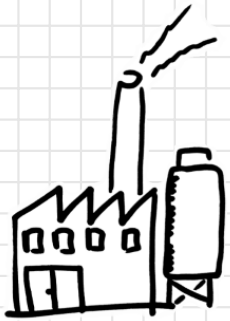
Open Code

Open Code

Data step 1

Data step 2

Data step 1



Proc FCMP and function
with Hash Table

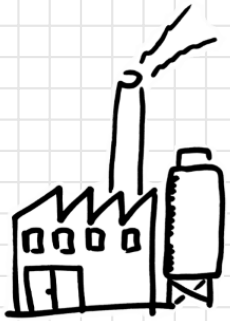
Open Code

Data step 2

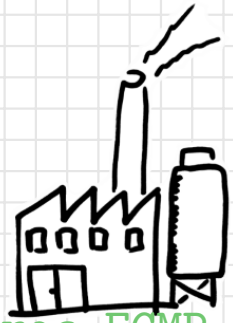
Open Code

Data step 1

Data step 2



Proc FCMP and function
with Hash Table



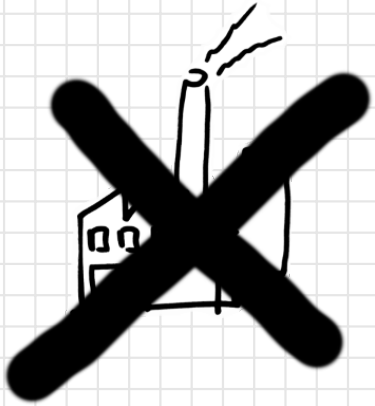
Proc FCMP and function
with Hash Table

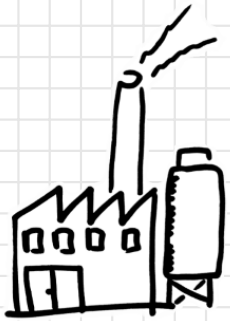
Open Code

Data step 1

Data step 2

FCMP+Resolve(%Macro)



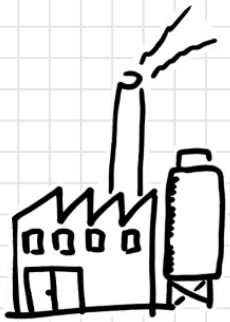


Proc FCMP and function
with Hash Table

Open Code

Data step 1

Data step 2



Proc FCMP and function
with Hash Table

FCMP+Resolve(%Macro) is a "bridge" which allows to exchange data between datasteps boundaries, inside a hash table embedded into a function, without reading the data from a disk drive each time.

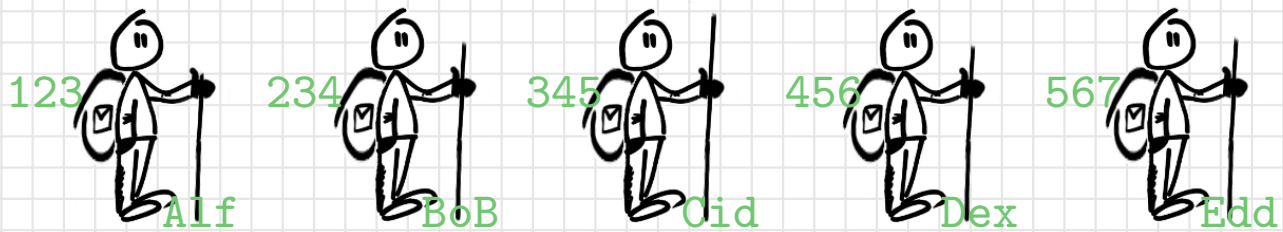
Art Carpenter introduced the idea of embedding FCMP with Hash table inside %MACRO to retrieve data.

Idea to use the RESOLVE() function to keep the same Hash Table between datasteps boundaries was independently suggested to Art by me and Thomas Billings.

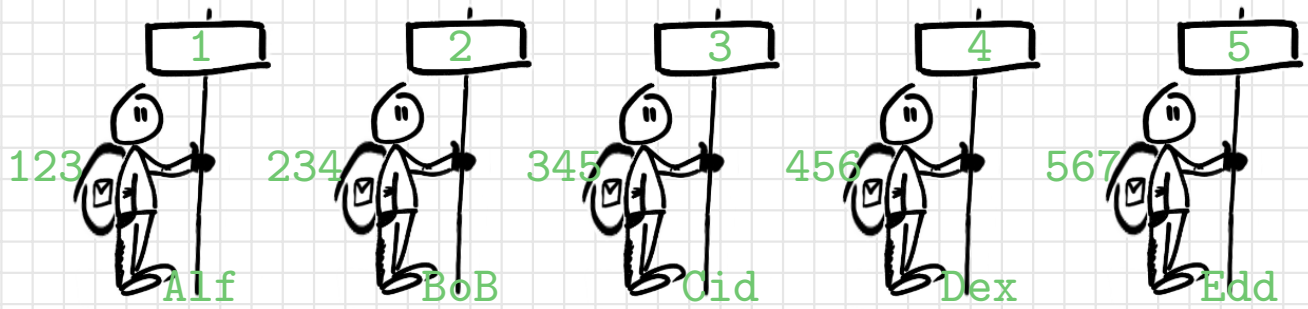
ii2#t

(Informal) Introduction to Hash Tables

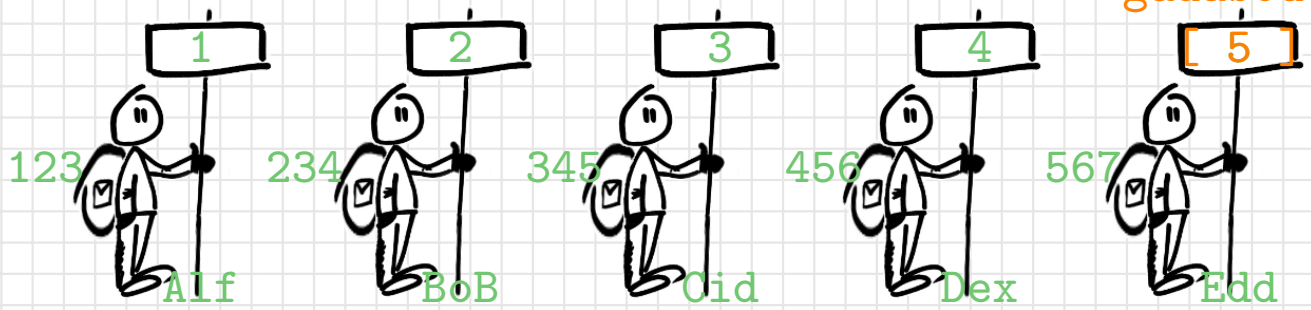
```
Alf = 123; BoB = 234; Cid = 345; Dex = 456; Edd = 567;
```



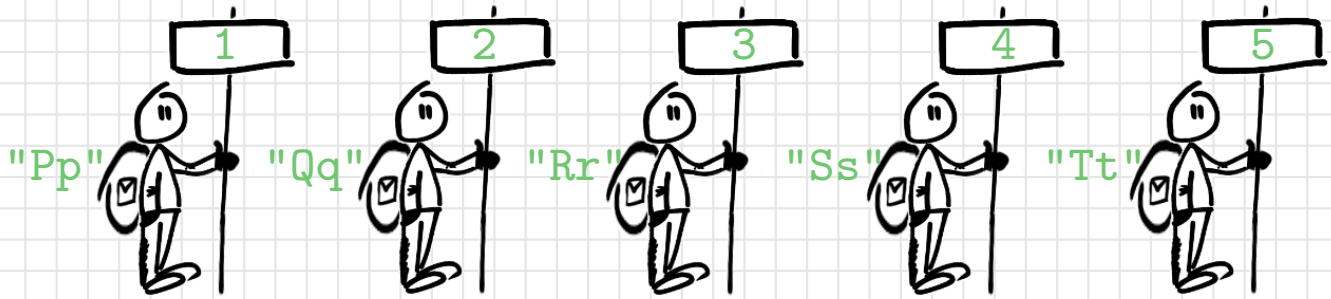
```
Alf = 123; BoB = 234; Cid = 345; Dex = 456; Edd = 567;

     ARRAY gadabouts[1:5] Alf BoB Cid Dex Edd;
```



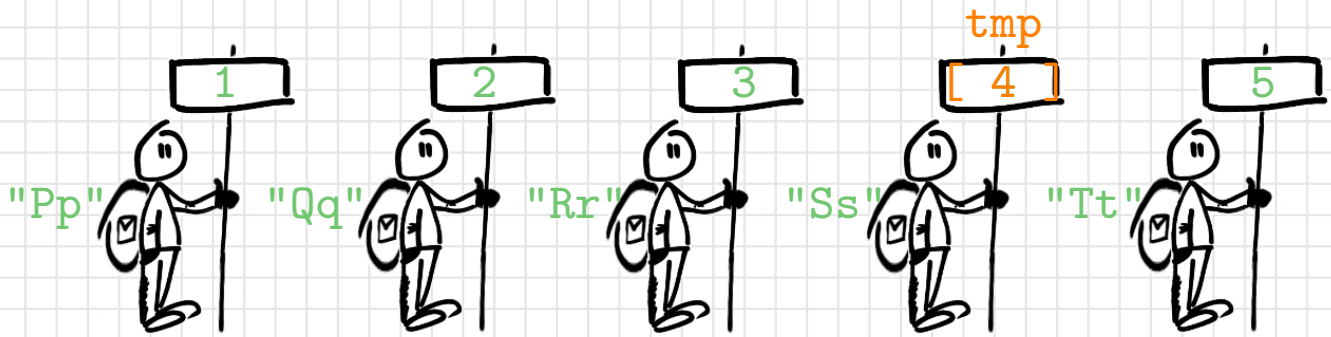
```
Alf = 123; BoB = 234; Cid = 345; Dex = 456; Edd = 567;

      ARRAY gadabouts[1:5] Alf BoB Cid Dex Edd;

      i = 5; PUT gadabouts[i]=; name = VNAME(gadabouts[i]);
```



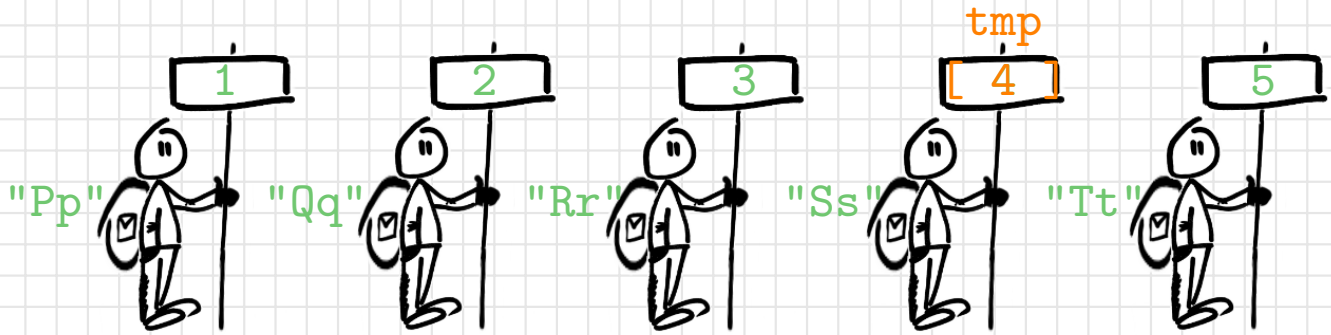
```
ARRAY tmp[1:5] $ 2 _TEMPORARY_ ("Pp" "Qq" "Rr" "Ss" "Tt");
```



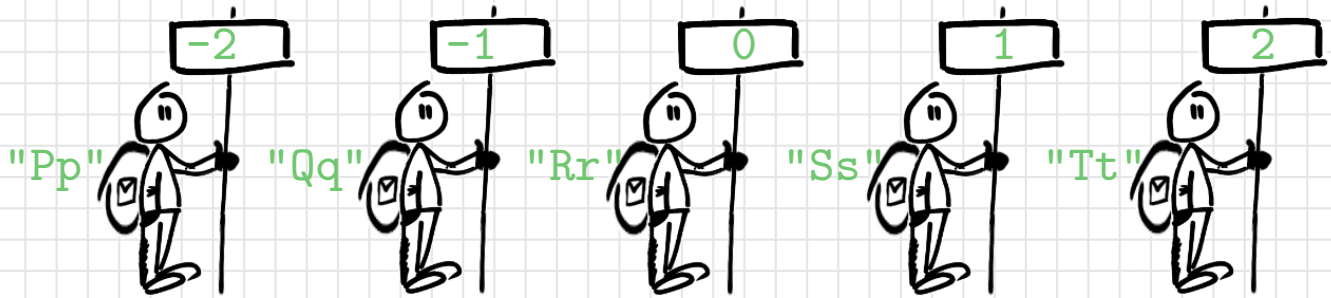
```
ARRAY tmp[1:5] $ 2 _TEMPORARY_ ("Pp" "Qq" "Rr" "Ss" "Tt");

      i = 4; put tmp[i]=;
```



```
ARRAY tmp[1:5] $ 2 _TEMPORARY_ ("Pp" "Qq" "Rr" "Ss" "Tt");

i = 4; put tmp[i]=; name = VNAME(tmp[i]);
```

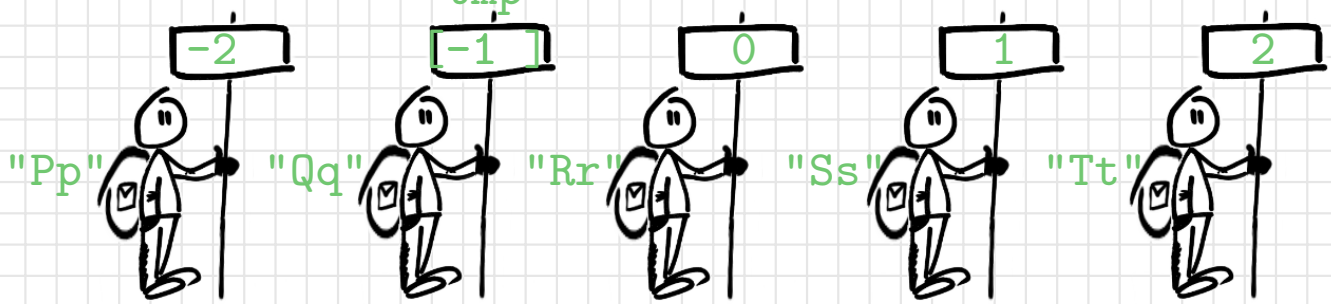


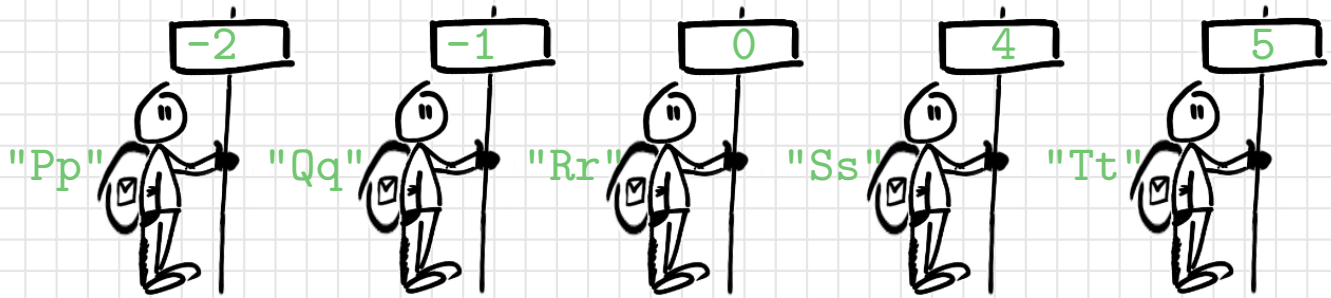

```
ERROR: Array elements of _TEMPORARY_ arrays are not defined
       as variables and therefore do NOT have NAMES or LABELS.
```

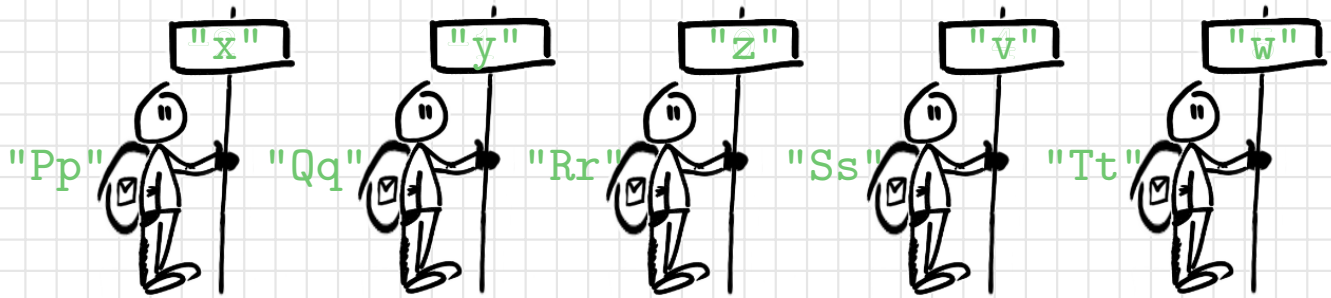

-2
-1
0
1
2
"Pp"
"Qq"
"Rr"
"Ss"
"Tt"

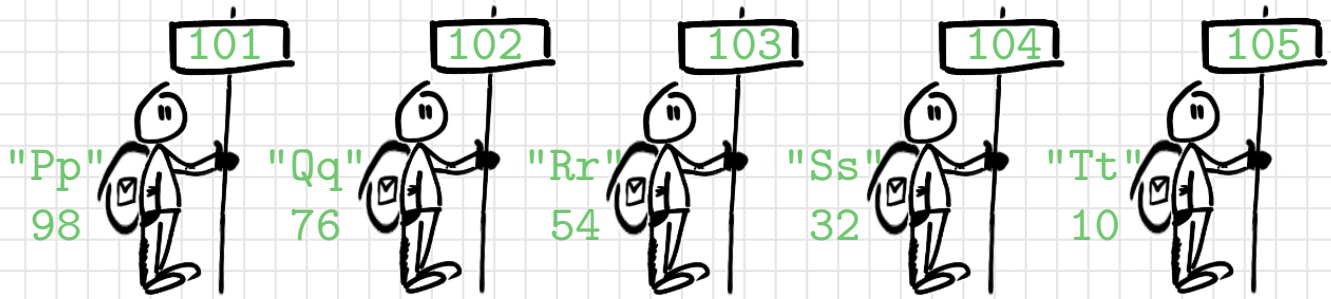
```
# OK #
```

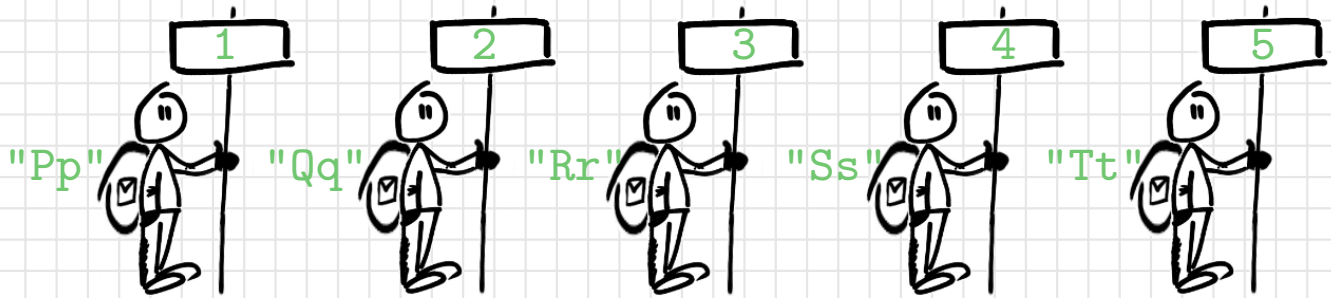
```
ARRAY tmp[-2:2] $ 2 _TEMPORARY_ ("Pp" "Qq" "Rr" "Ss" "Tt");

        i =-1; put tmp[i]=;
```




-2
-1
0
4
5
"Pp"
"Qq"
"Rr"
"Ss"
"Tt"


"x"
"y"
"z"
"v"
"w"
"Pp"
"Qq"
"Rr"
"Ss"
"Tt"


101
102
103
104
105
"Pp"
98
"Qq"
76
"Rr"
54
"Ss"
32
"Tt"
10

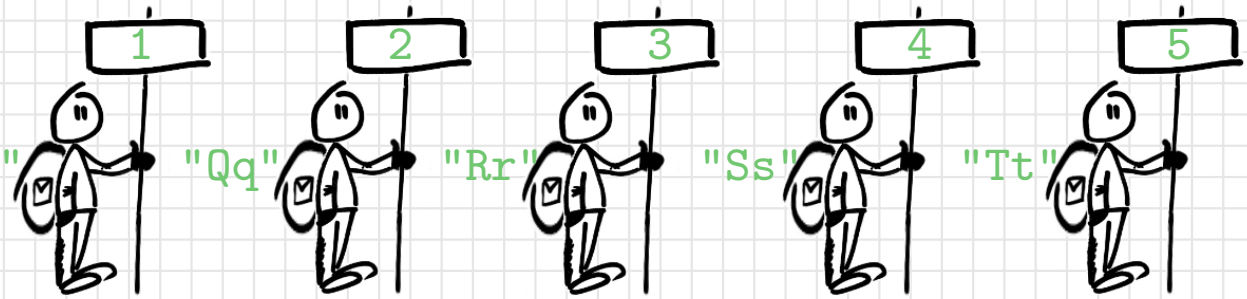

1
2
3
4
5
"Pp"
"Qq"
"Rr"
"Ss"
"Tt"

hash world

```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```

array world(kind of)

```
ARRAY a
[1:5]
$ 2 _temporary_
;
```



add data:

```
k=1; d="Pp"; h.ADD();
```

```
k=2; d="Qq"; a[k] = d;
```

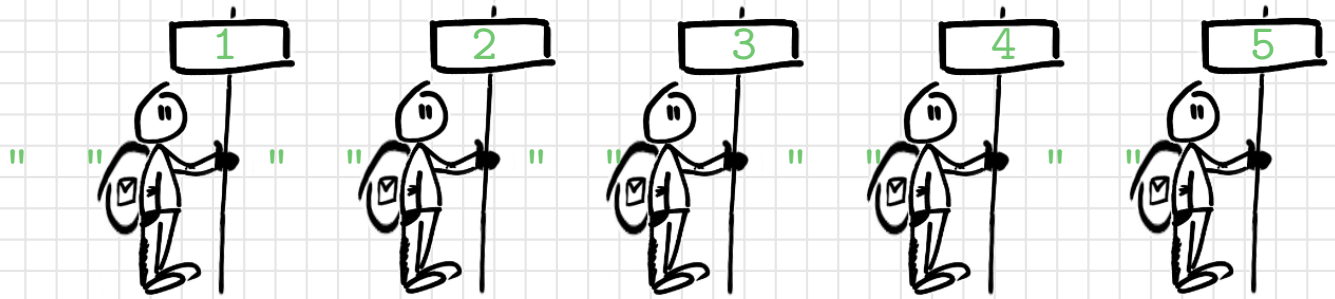
get data:

```
k=3; h.FIND(); put d=;
```

```
k=4; d = a[k]; put d=;
```

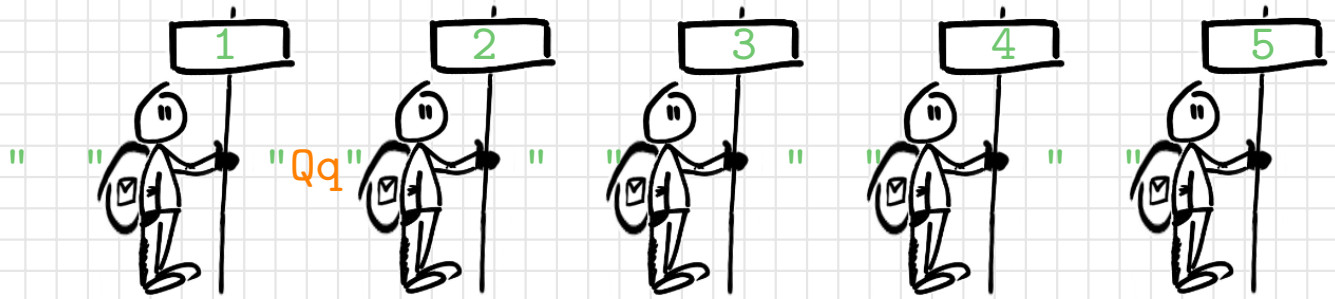
temporary array "vs." hash table

```
ARRAY a
[1:5]
$ 2 _temporary_
;
```



temporary array "vs." hash table

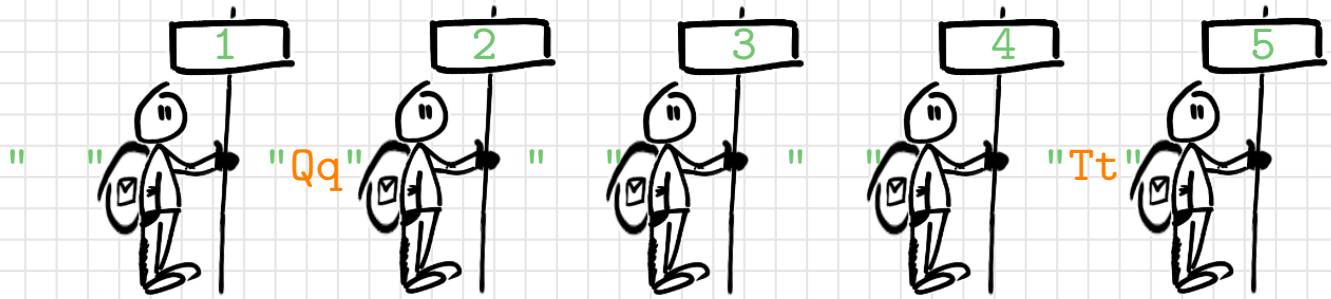
```
ARRAY a
[1:5]
$ 2 _temporary_
;
```




```
k=2; d="Qq"; a[k] = d;
```

temporary array "vs." hash table

```
ARRAY a
[1:5]
$ 2 _temporary_
;
```

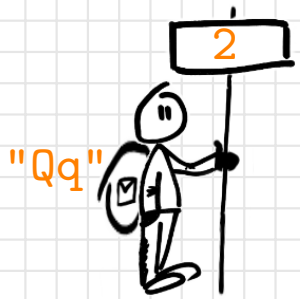



```
k=2; d="Qq"; a[k] = d;
k=5; d="Tt"; a[k] = d;
...
..
.
```

temporary array "vs." hash table

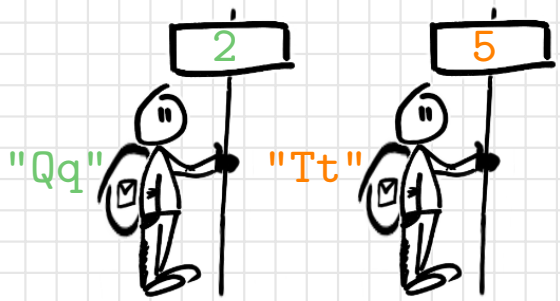
```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```

```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```



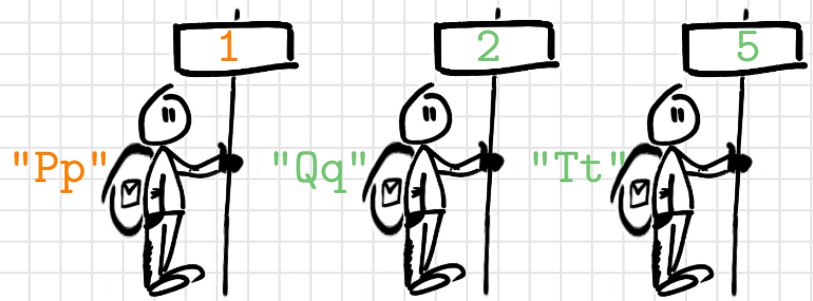

```
k=2; d="Qq"; h.ADD();
```

```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```



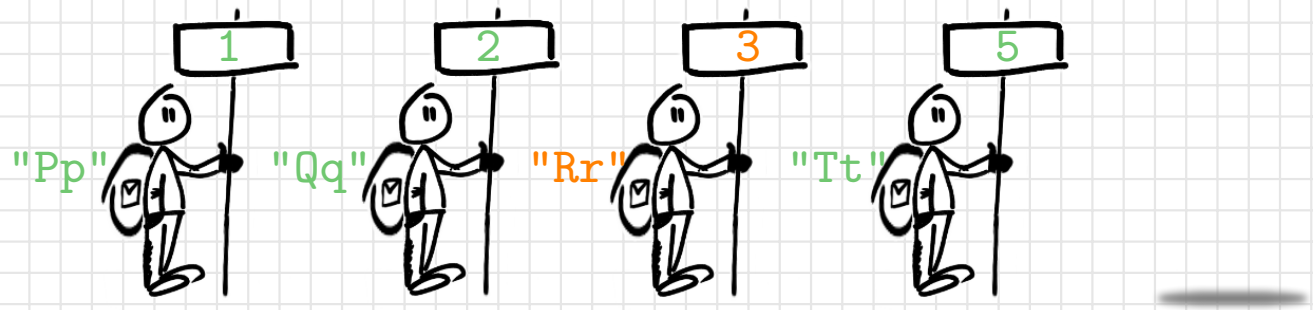

```
k=2; d="Qq"; h.ADD();
k=5; d="Tt"; h.ADD();
```

```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```



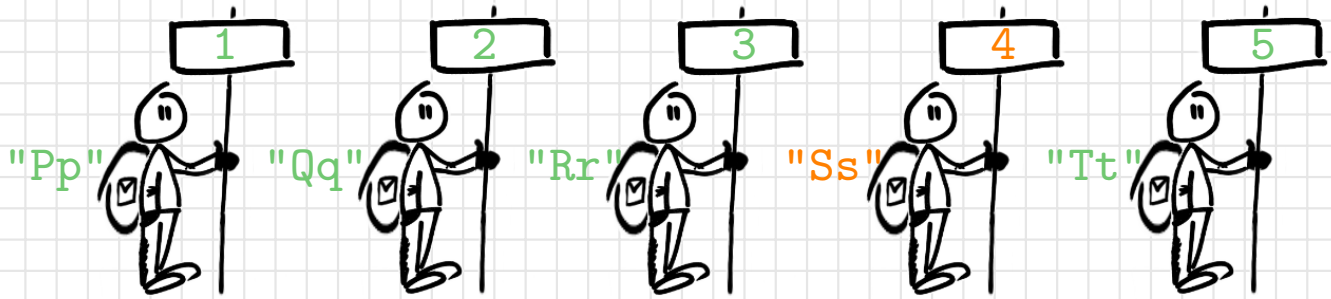

```
k=2; d="Qq"; h.ADD();
k=5; d="Tt"; h.ADD();
k=1; d="Pp"; h.ADD();
```

```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```



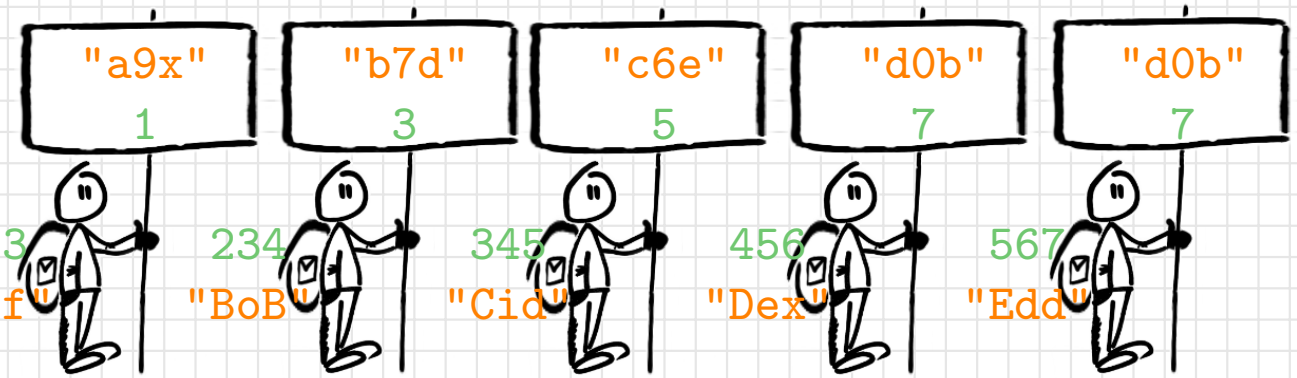

```
k=2; d="Qq"; h.ADD();
k=5; d="Tt"; h.ADD();
k=1; d="Pp"; h.ADD();
k=3; d="Rr"; h.ADD();
```

```
LENGTH k 8 d $ 2;
DECLARE HASH h(ordered:"a");
h.DefineKey("k");
h.DefineData("d");
h.DefineDone();
```




```
k=2; d="Qq"; h.ADD();
k=5; d="Tt"; h.ADD();
k=1; d="Pp"; h.ADD();
k=3; d="Rr"; h.ADD();
k=4; d="Ss"; h.ADD();
```

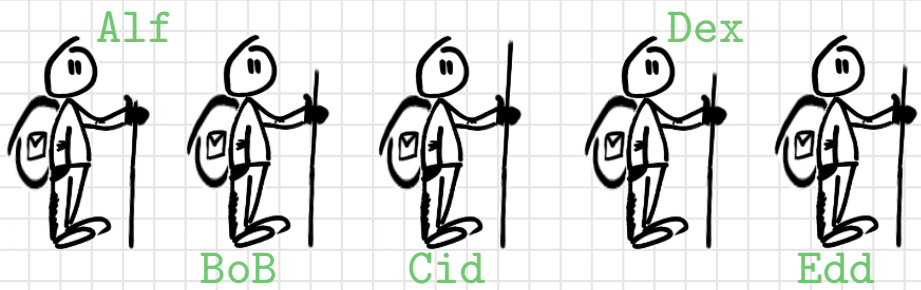
```
IF 0 THEN SET work.dataset;
DECLARE HASH h(dataset: "work.dataset");
h.DefineKey("cK", "nK");
h.DefineData("nD", "cD");
h.DefineDone();
```

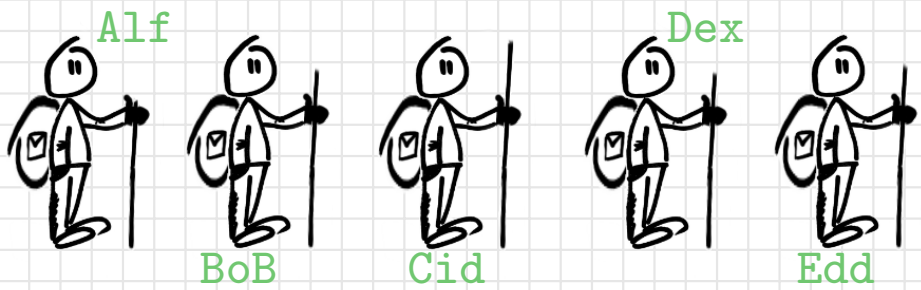



```
...
STOP;
RUN;
```

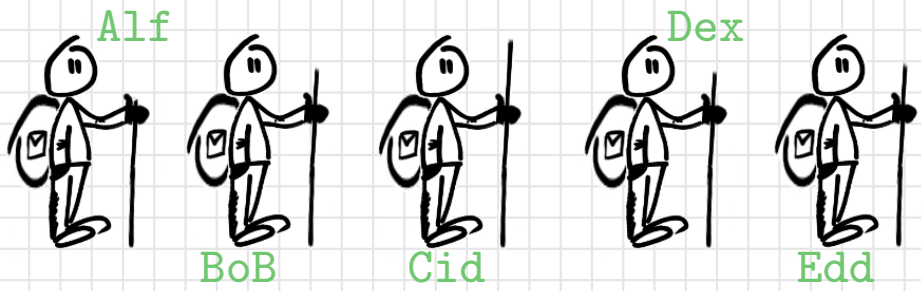
ii2udf&s

(Informal) Introduction to User Defined Functions & Subroutines

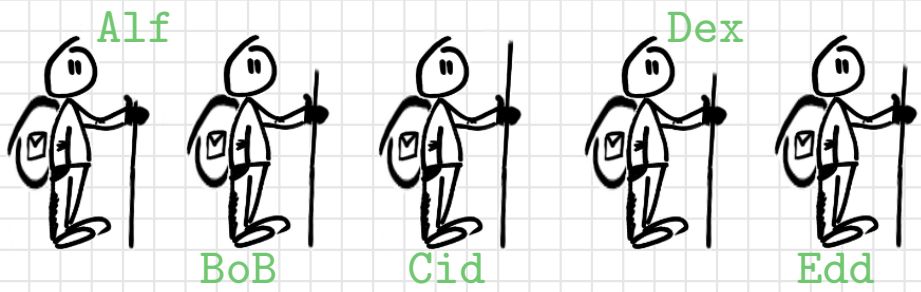

Alf
BoB
Cid
Dex
Edd



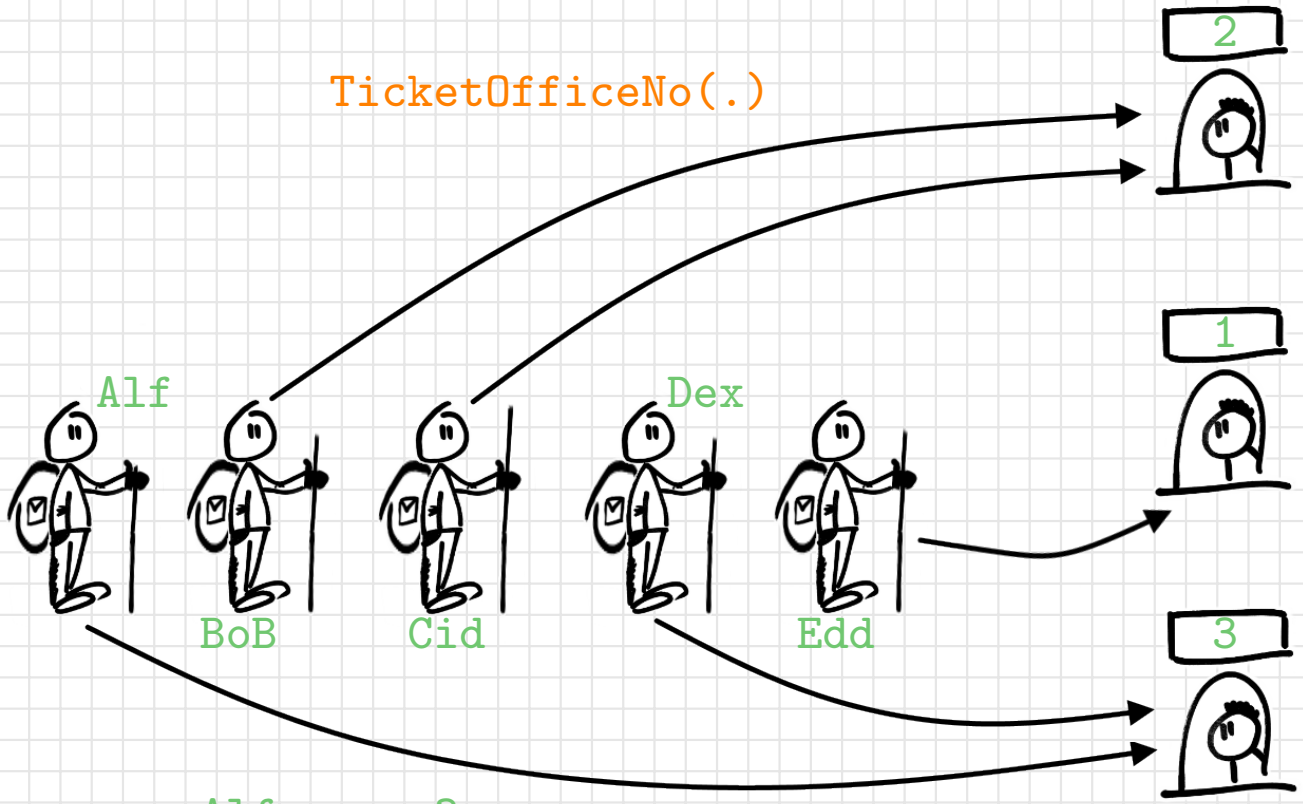

```
Alf
BoB
Cid
Dex
Edd
```



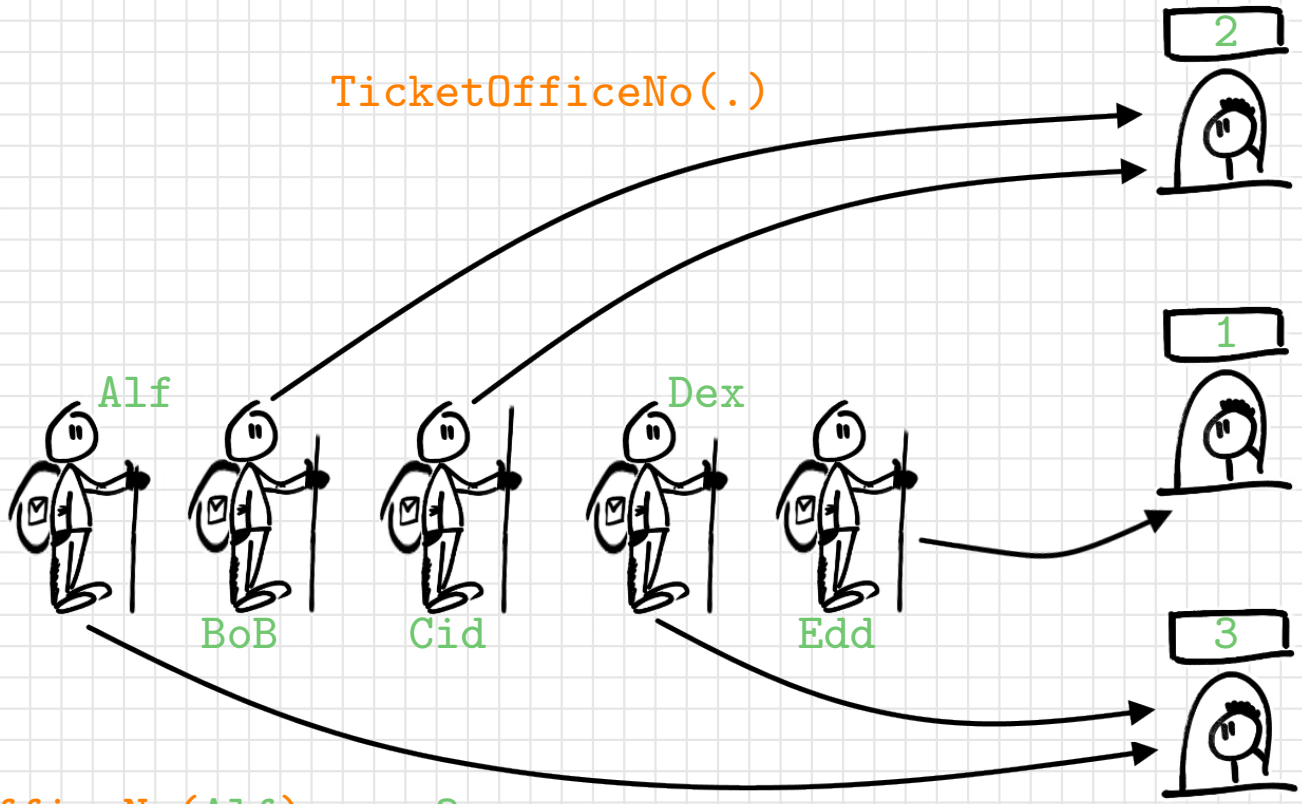

```
Alf
BoB
Cid
Dex
Edd
```



Alf	3
BoB	
Cid	2
Dex	
Edd	1



Alf	3
BoB	
Cid	2
Dex	
Edd	1




```
TicketOfficeNo(Alf)  =  3
TicketOfficeNo(BoB)  =  2
TicketOfficeNo(Cid)  =  2
TicketOfficeNo(Dex)  =  3
TicketOfficeNo(Edd)  =  1
```

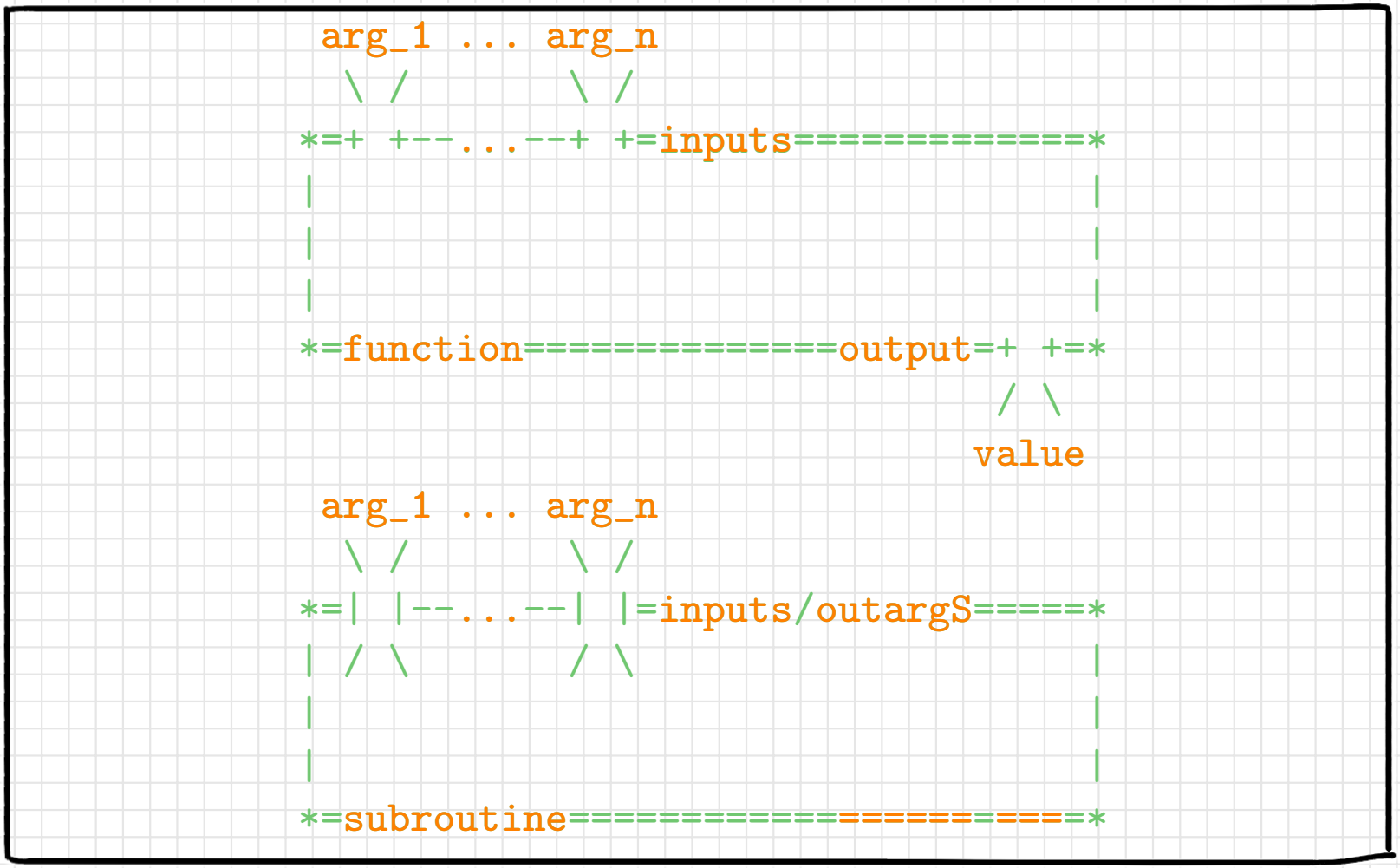
```
   PROC FCMP;                                                    #
01  FUNCTION num_fun(num_arg, char_arg $);
02   ...
03   RETURN(num_value);
04  ENDSUB;
05
06   FUNCTION char_fun(num_arg, char_arg $) $;
07    ...
08    RETURN(char_value);
09   ENDSUB;
10
11    FUNCTION array_arg_fun(array_arg[*]) VARARGS;
12     ...
13     RETURN(one_value);
14    ENDSUB;
   RUN;                                                          #
```

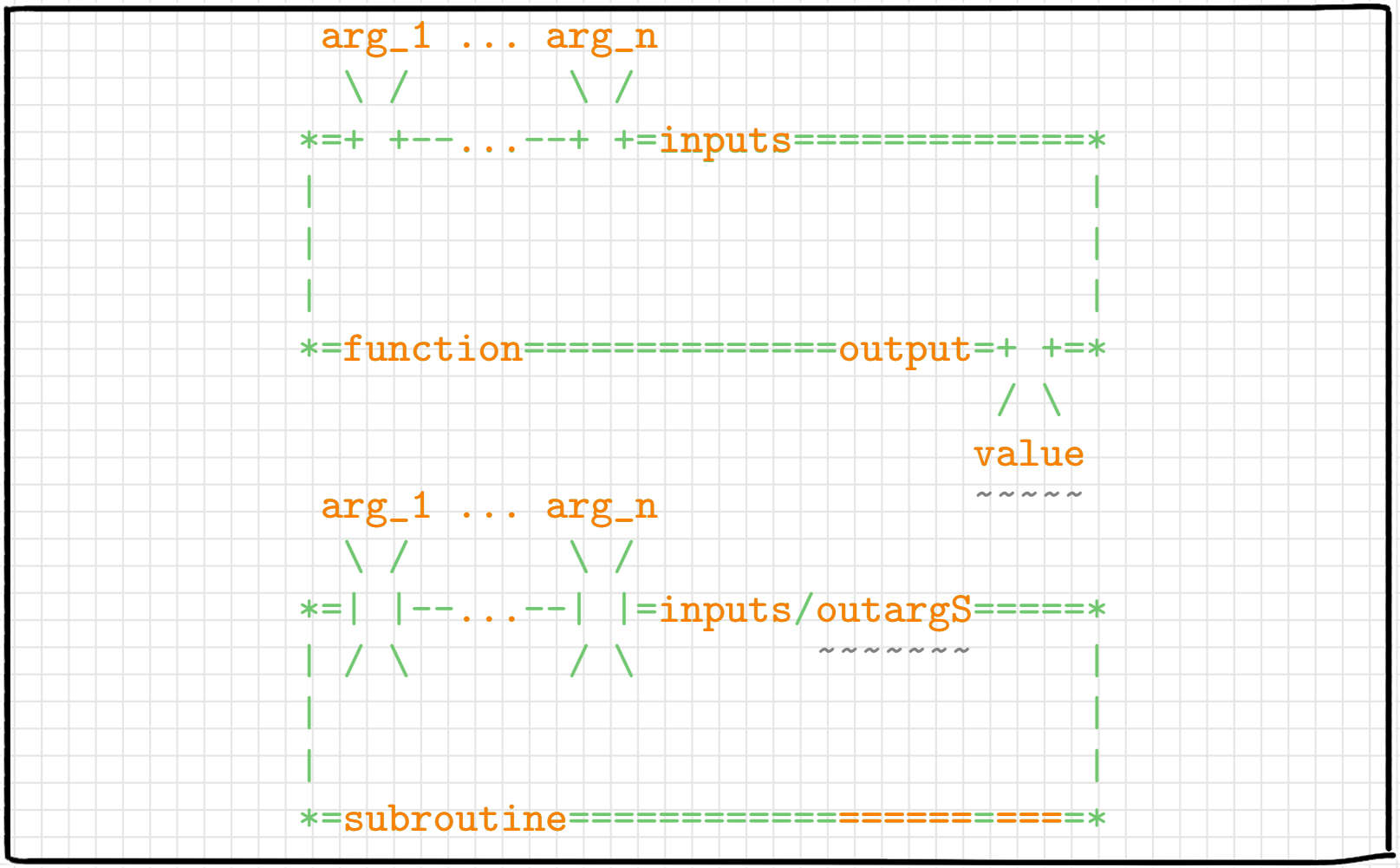
```
   PROC FCMP                                                    #
01  OUTLIB = library.dataset.package;
02   FUNCTION Pythagoras(a, b);
03    c = SQRT(a**2 + b**2);
04    RETURN(c);
05   ENDSUB;
06 RUN;
07
08 OPTIONS CMPLIB = (library.dataset _DISPLAYLOC_);
09
10 DATA _null_;
11  a = 3;
12  b = 4;
13  c = Pythagoras(a, b);
14  PUT _ALL_;
   RUN;                                                         #
```

```
   PROC FCMP                                                    #
01  OUTLIB = library.dataset.package;
02   FUNCTION Pythagoras(a, b);
03    c = SQRT(a**2 + b**2);
04    RETURN(c);
05   ENDSUB;
06 RUN;
07
08 OPTIONS CMPLIB = (library.dataset _DISPLAYLOC_);
09
10 DATA _null_;
11  a = 3;
12  b = 4;
13  c = Pythagoras(a, b);
14  PUT _ALL_;
   RUN;                                                         #
```

```
NOTE: Function 'Pythagoras' loaded from library.dataset.PACKAGE.
a=3 b=4 c=5 _ERROR_=0 _N_=1
```

```
PROC FCMP;                                                              #
 SUBROUTINE arg3out2sub(first_arg, second_arg, third_arg);
  OUTARGS first_arg, third_arg;
  ...
  RETURN;
 ENDSUB;

  SUBROUTINE subarray(arrone[*], arrtwo[*], arrthree[*]);
   OUTARGS arrthree;
   DO _i_ = 1 TO DIM(arrthree);
    arrthree[_i_] = arrone[_i_]**2 + arrtwo[_i_]**2;
   END;
   RETURN;
  ENDSUB;

RUN;                                                                    #
```


arg_1 ... arg_n
inputs
function
output
value
arg_1 ... arg_n
inputs/outargS
subroutine


arg_1 ... arg_n
inputs
function
output
value
arg_1 ... arg_n
inputs/outargS
subroutine

Function[★] is a set F of ordered pairs (*argument*, *value*) such that for each *argument* there exists exactly one *value*.

[★] Remember title of this section

Function[★] is a set F of ordered pairs (*argument*, *value*) such that for each *argument* there exists exactly one *value*.

example[*]:

subject_id	(trtstdt, trtendt)
001	(2018-01-01, 2019-01-01)
002	(2018-02-01, 2019-01-02)
003	(2018-03-01, 2019-01-03)
004	(2018-02-01, 2019-01-02)
005	(2018-05-01, 2019-01-05)

[*] tricky

[★] Remember title of this section

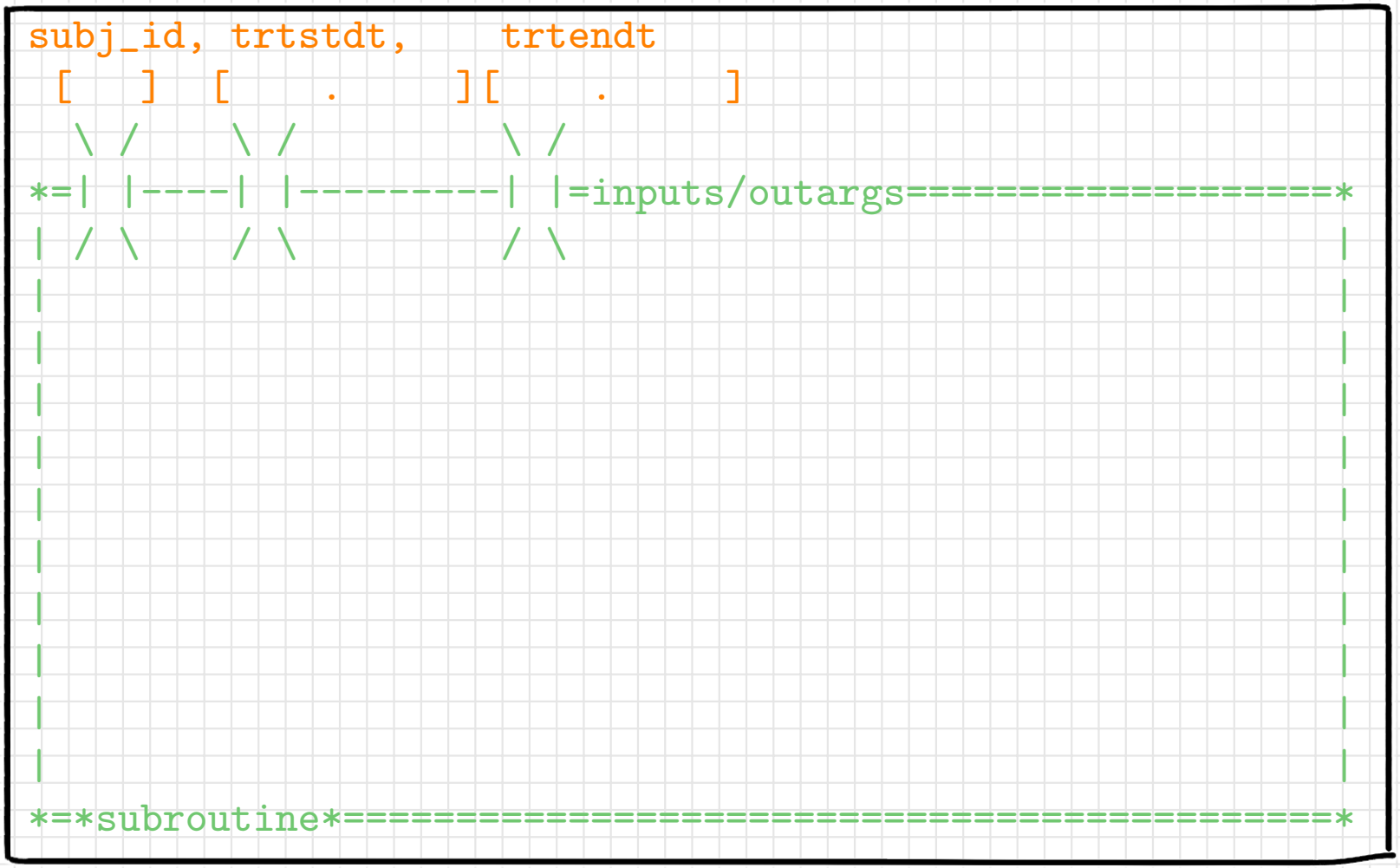
Function[★] is a set F of ordered pairs (*argument*, *value*) such that for each *argument* there exists exactly one *value*.

counter-example:

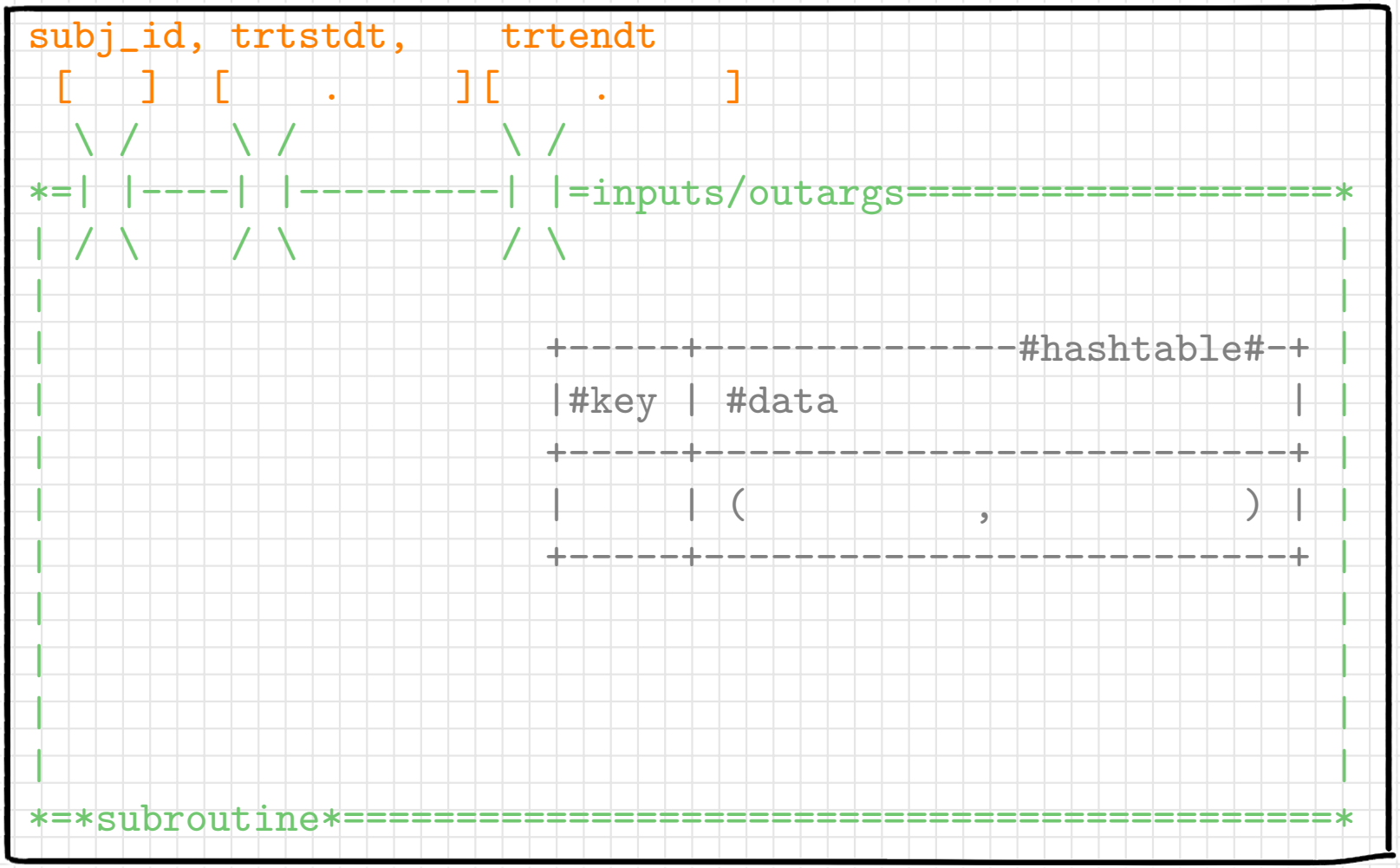
subject_id	subject_name
001	Alf
002	BoB
003	Cid
002	Dex
005	Edd

[★] Remember title of this section

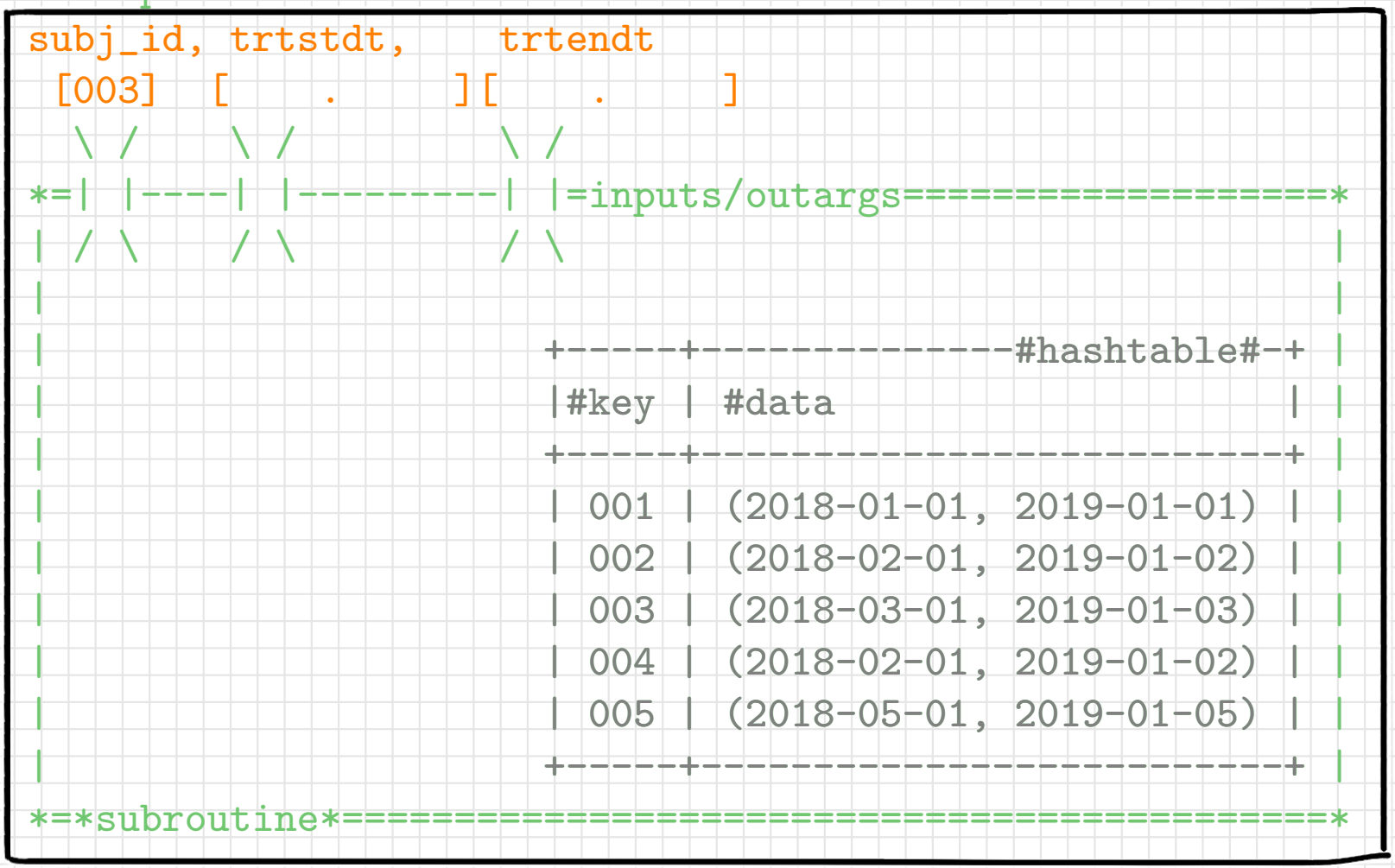
Proc FCMP:



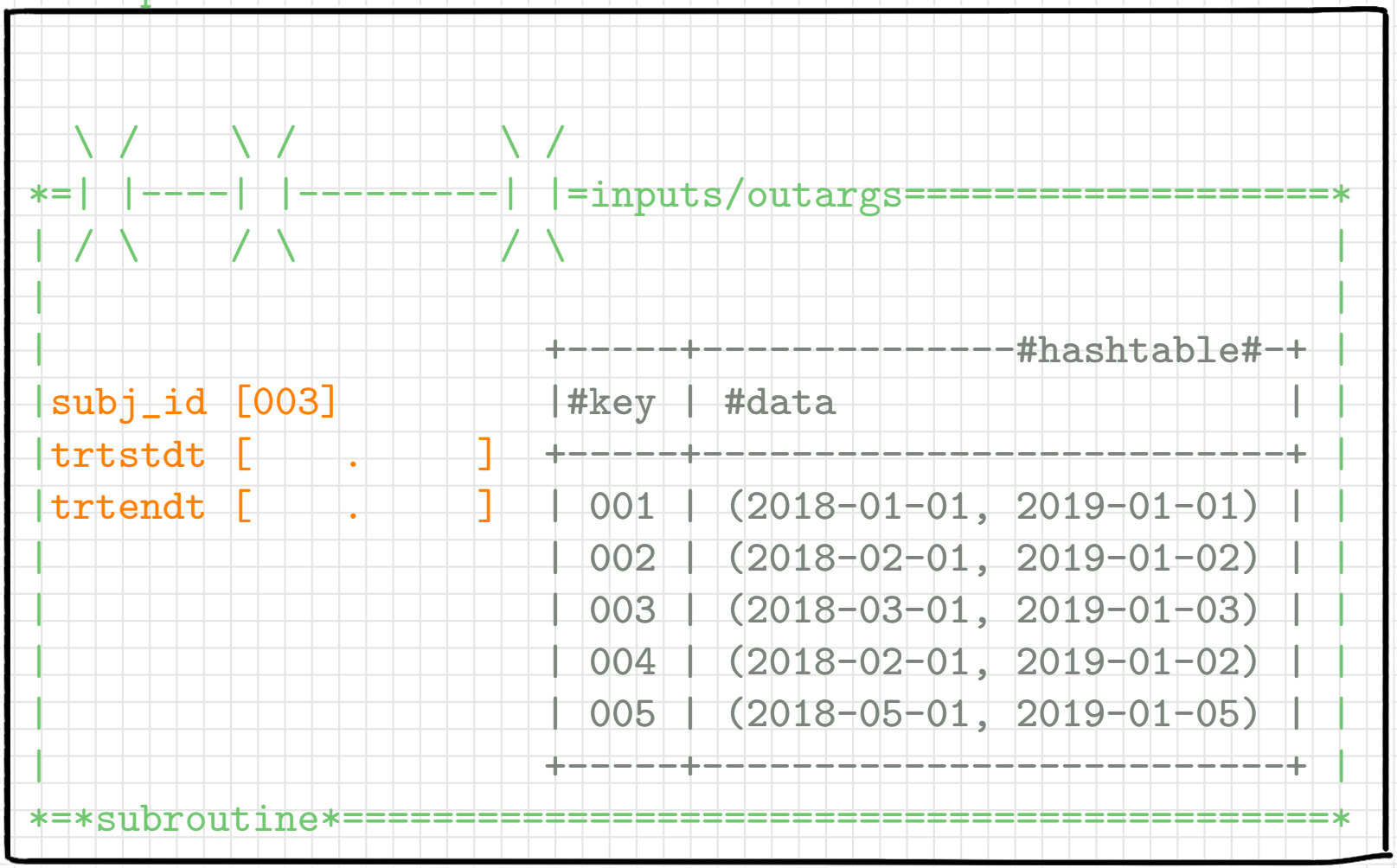
Proc FCMP:


subj_id, trtstdt, trtendt
inputs/outargs
#hashtable#
#key
#data
(,)
subroutine

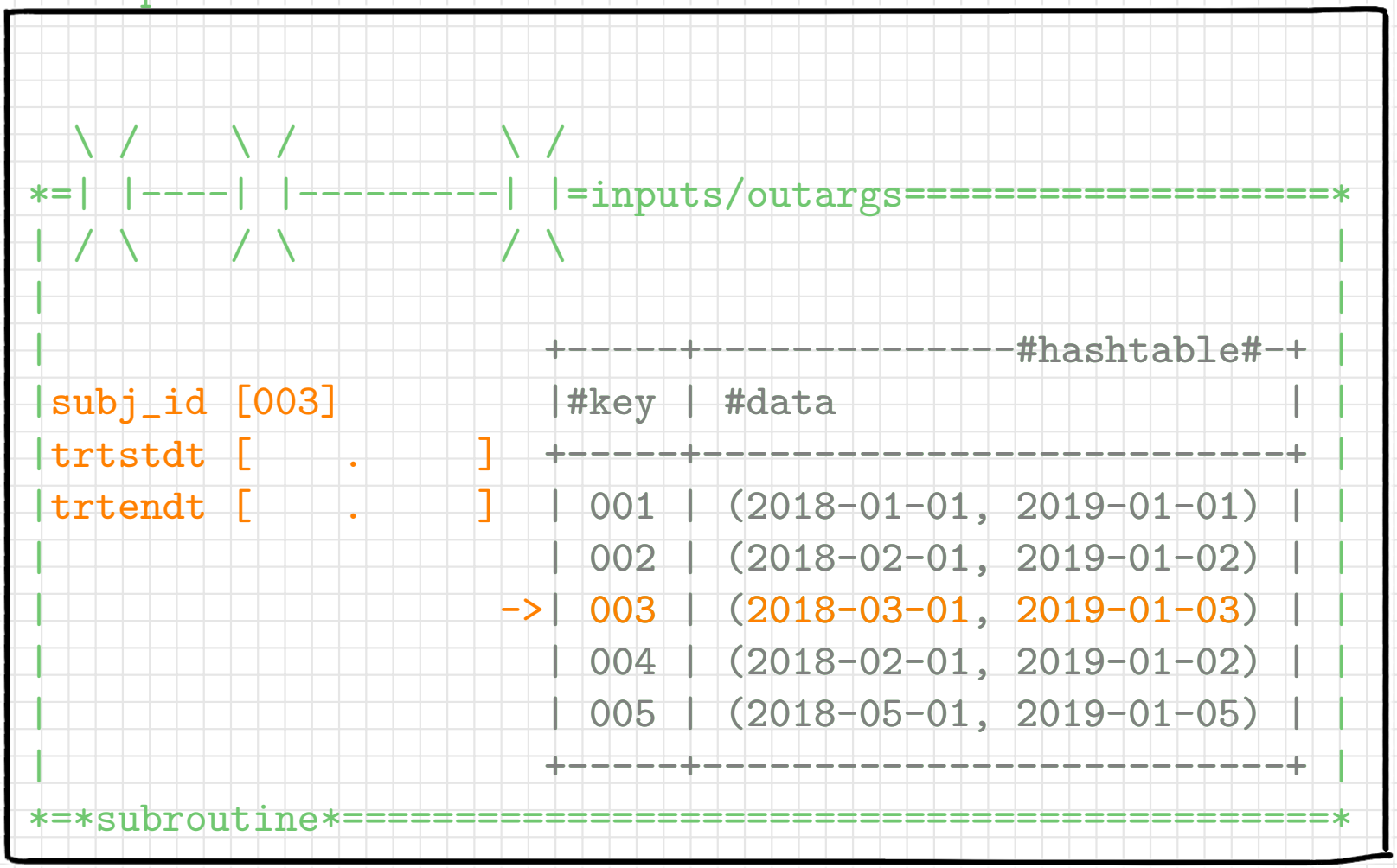
Data Step:


subj_id, trtstdt, trtendt
[003] [.][.]
inputs/outargs
#hashtable#
#key | #data
001 | (2018-01-01, 2019-01-01)
002 | (2018-02-01, 2019-01-02)
003 | (2018-03-01, 2019-01-03)
004 | (2018-02-01, 2019-01-02)
005 | (2018-05-01, 2019-01-05)
subroutine

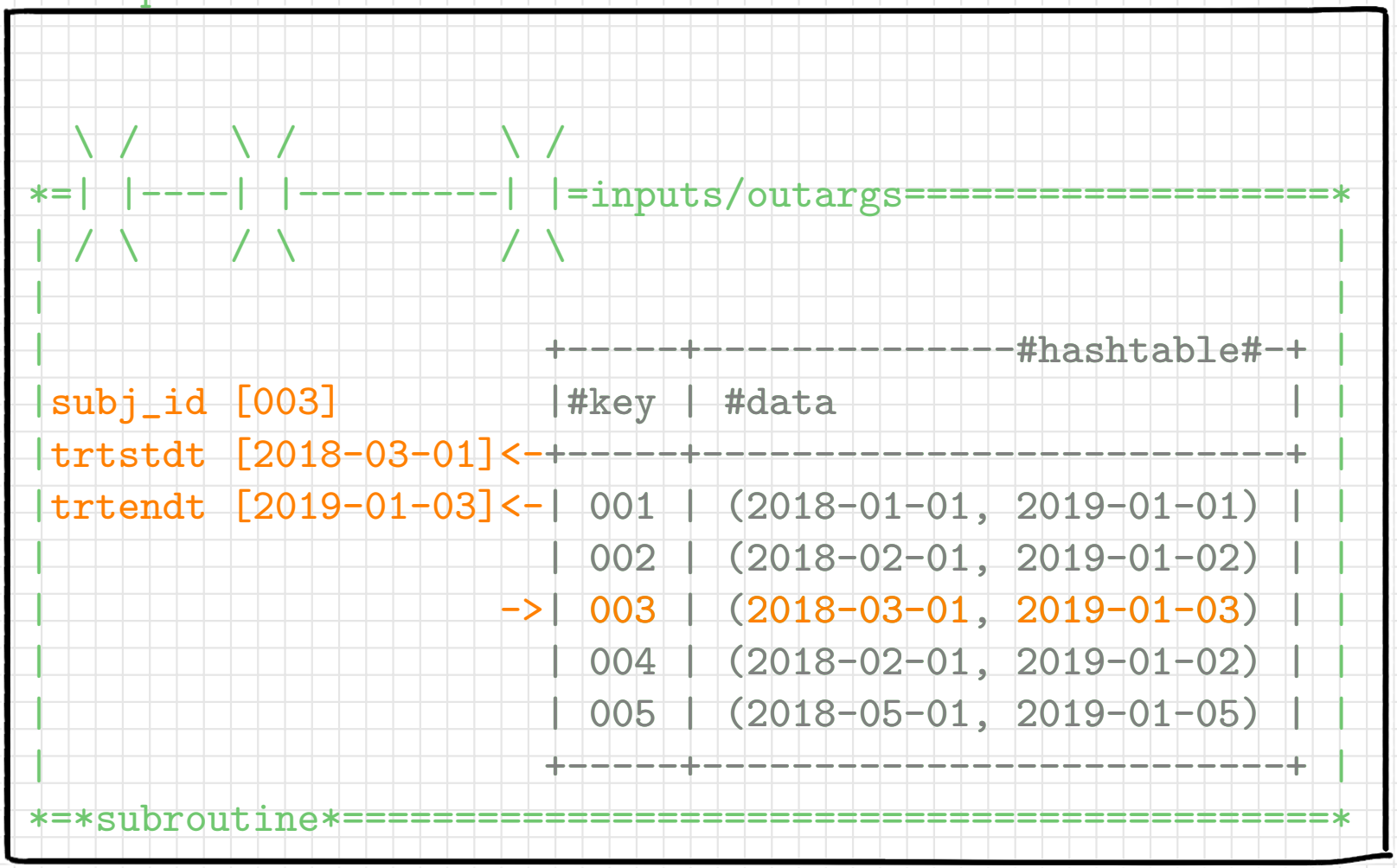
Data Step:


=|
=inputs/outargs=
subj_id [003]
trtstdt [.]
trtendt [.]
#hashtable#
#key | #data
001 | (2018-01-01, 2019-01-01)
002 | (2018-02-01, 2019-01-02)
003 | (2018-03-01, 2019-01-03)
004 | (2018-02-01, 2019-01-02)
005 | (2018-05-01, 2019-01-05)
*=*subroutine*

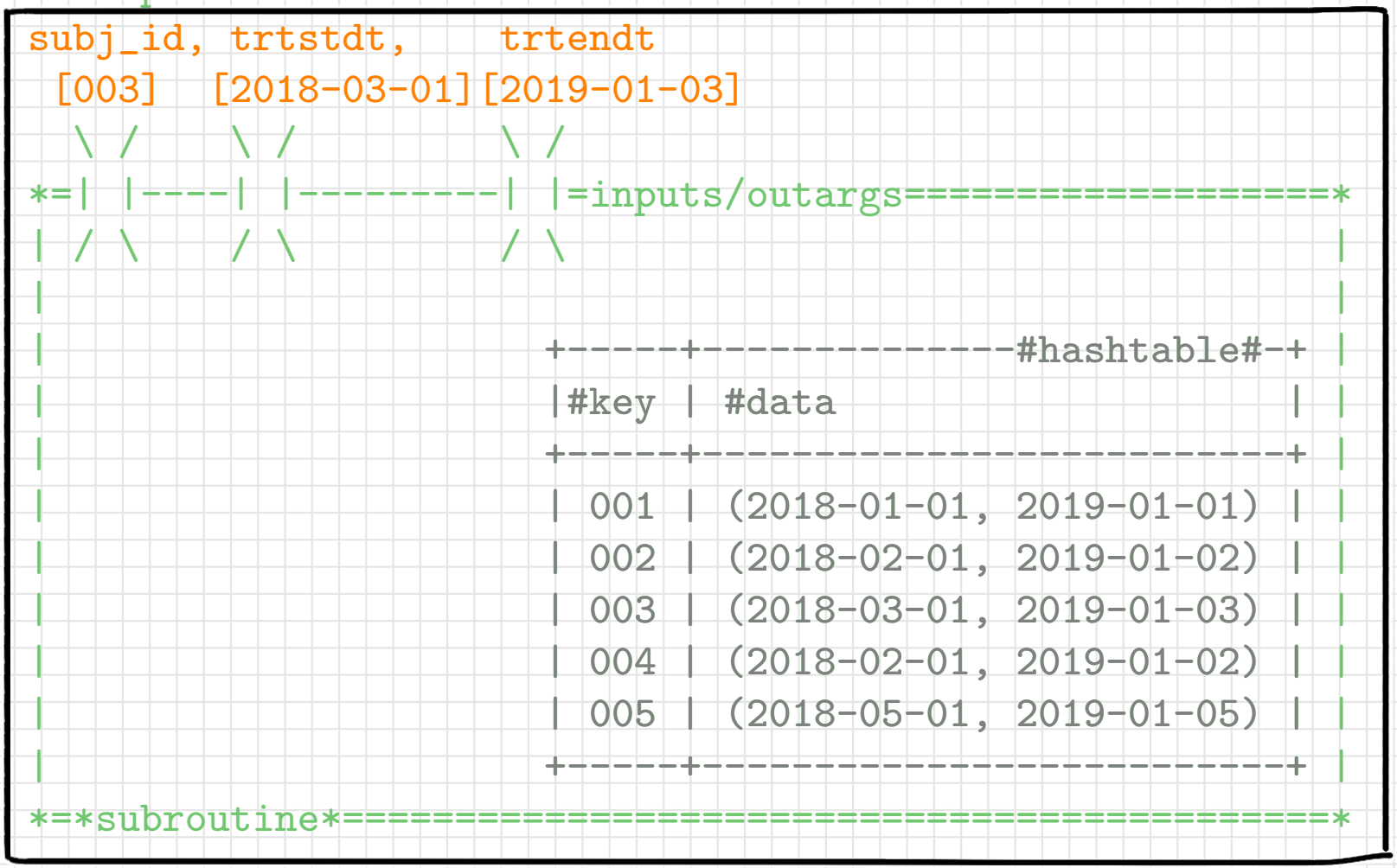
Data Step:

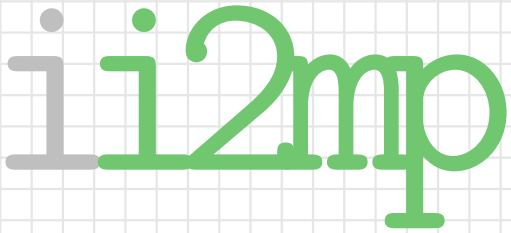

=inputs/outargs
subj_id [003]
trtstdt [.]
trtendt [.]
#hashtable#
#key | #data
001 | (2018-01-01, 2019-01-01)
002 | (2018-02-01, 2019-01-02)
003 | (2018-03-01, 2019-01-03)
004 | (2018-02-01, 2019-01-02)
005 | (2018-05-01, 2019-01-05)
subroutine

Data Step:


inputs/outargs
subj_id [003]
trtstdt [2018-03-01]
trtendt [2019-01-03]
#hashtable#
#key | #data
001 | (2018-01-01, 2019-01-01)
002 | (2018-02-01, 2019-01-02)
003 | (2018-03-01, 2019-01-03)
004 | (2018-02-01, 2019-01-02)
005 | (2018-05-01, 2019-01-05)
subroutine

Data Step:


subj_id, trtstdt, trtendt
[003] [2018-03-01][2019-01-03]
inputs/outargs
#hashtable#
#key | #data
001 | (2018-01-01, 2019-01-01)
002 | (2018-02-01, 2019-01-02)
003 | (2018-03-01, 2019-01-03)
004 | (2018-02-01, 2019-01-02)
005 | (2018-05-01, 2019-01-05)
subroutine



(Informal) Introduction to Macro Programing

Data processing languages like:

`4GL`, `SQL`, `IML`, `DS2` or `SCL`

are SAS dialects (or slangs) which's data are data (e.g. SAS datasets, tables, matrices, databases or lists...)

While `MACRO` language is a SAS dialect which's data is... your code...

%

and &

%macroinstruction

and &macrovariable

What you run :

```
%MACRO abc(n, prefix=Set_Nmbr, distr=Normal);
DATA work.&prefix._&n;
 CALL STREAMINIT(&n);
 DO _N_ = 1 TO %SYSFUNC(SQRT(&n));
  x = RAND("&distr.") * &n;
  OUTPUT;
 END;
RUN;
%MEND abc;

%abc(42, distr=Uniform)
%abc(17, prefix=The_Best)
```

What SAS runs:

```
DATA work.Set_Nmbr_42;
 CALL STREAMINIT(42);
 DO _N_ = 1 TO 6.4807406984078602;
  x = RAND("Uniform") * 42;
  OUTPUT;
 END;
RUN;

DATA work.The_Best_17;
 CALL STREAMINIT(17);
 DO _N_ = 1 TO 4.1231056256176605;
  x = RAND("Normal ") * 17;
  OUTPUT;
 END;
RUN;
```

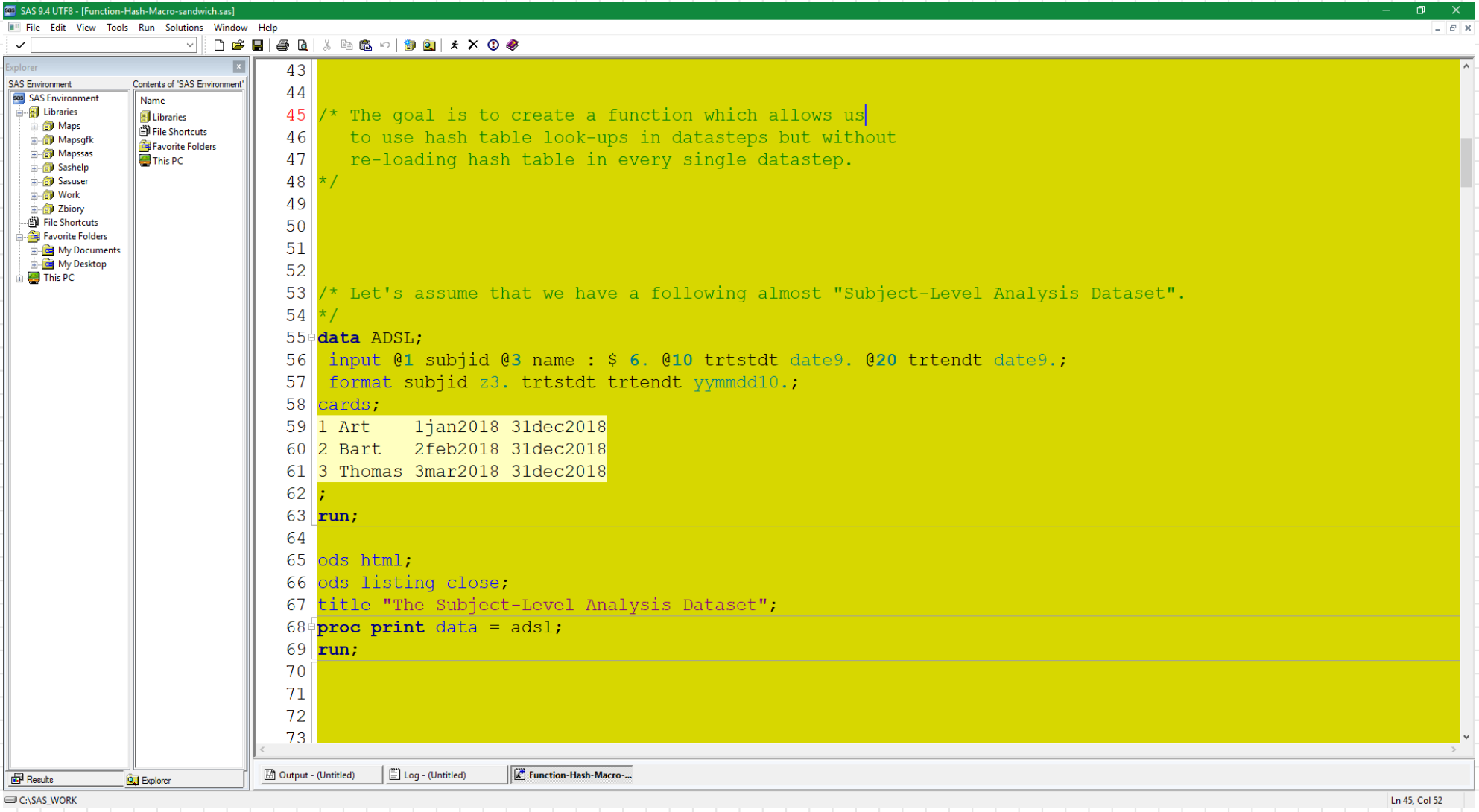
```
%LET arg1 = ... ;
  ...
%LET argN = ... ;
%LET val =
 %SYSFUNC(UsrDfFunc(&arg1,...,&argN));
%PUT &=val;

%LET ioarg1 = ... ;
  ...
%LET ioargN = ... ;
 %SYSCALL UsrDfSbrt(ioarg1,...,ioargN);
%PUT &=ioarg1 ... &=ioargN;
```

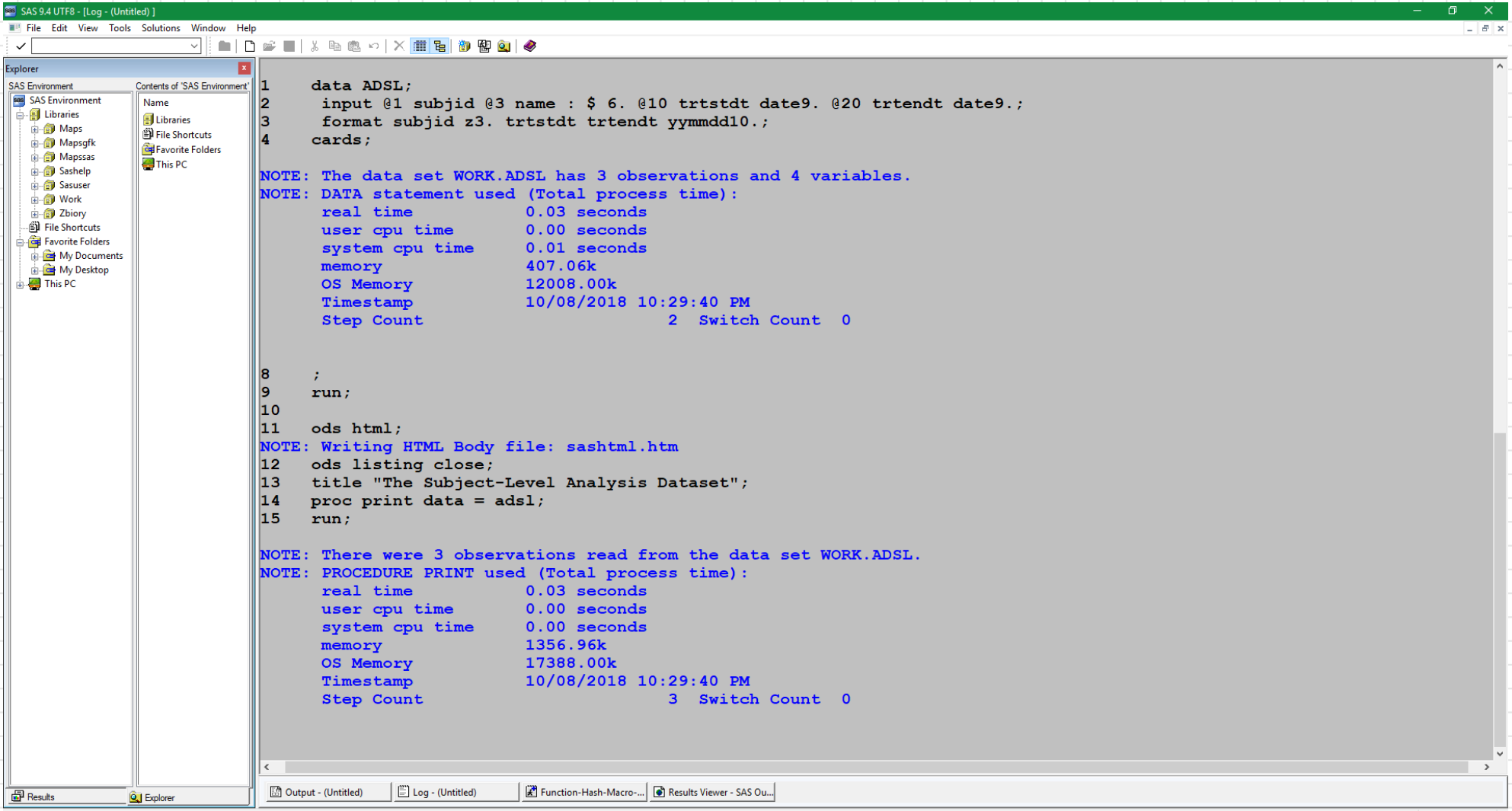
(Almost) Live Demo

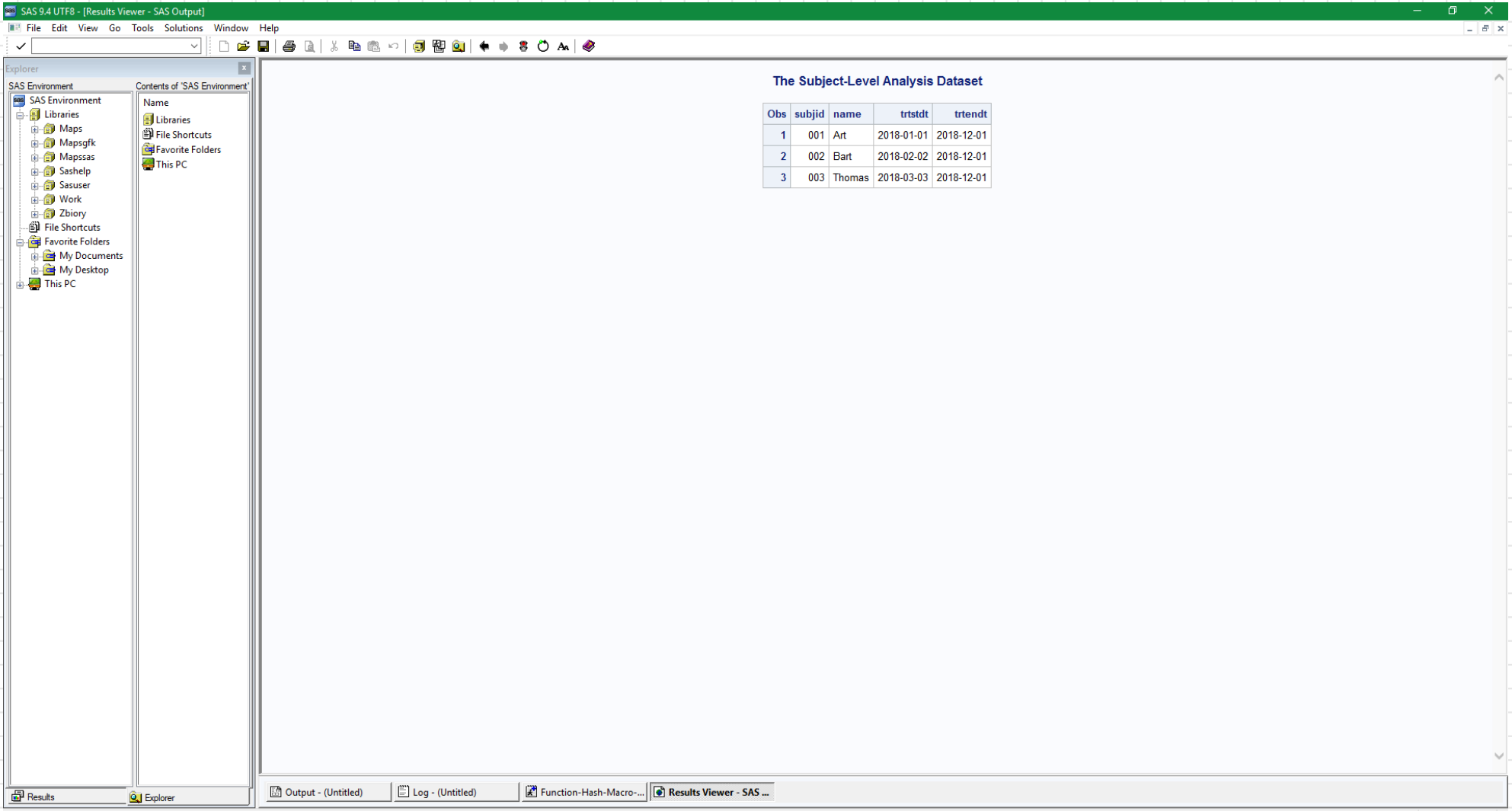
In fact it is a demo of a demo... go to the following web address and read the code to see the "bare metal".

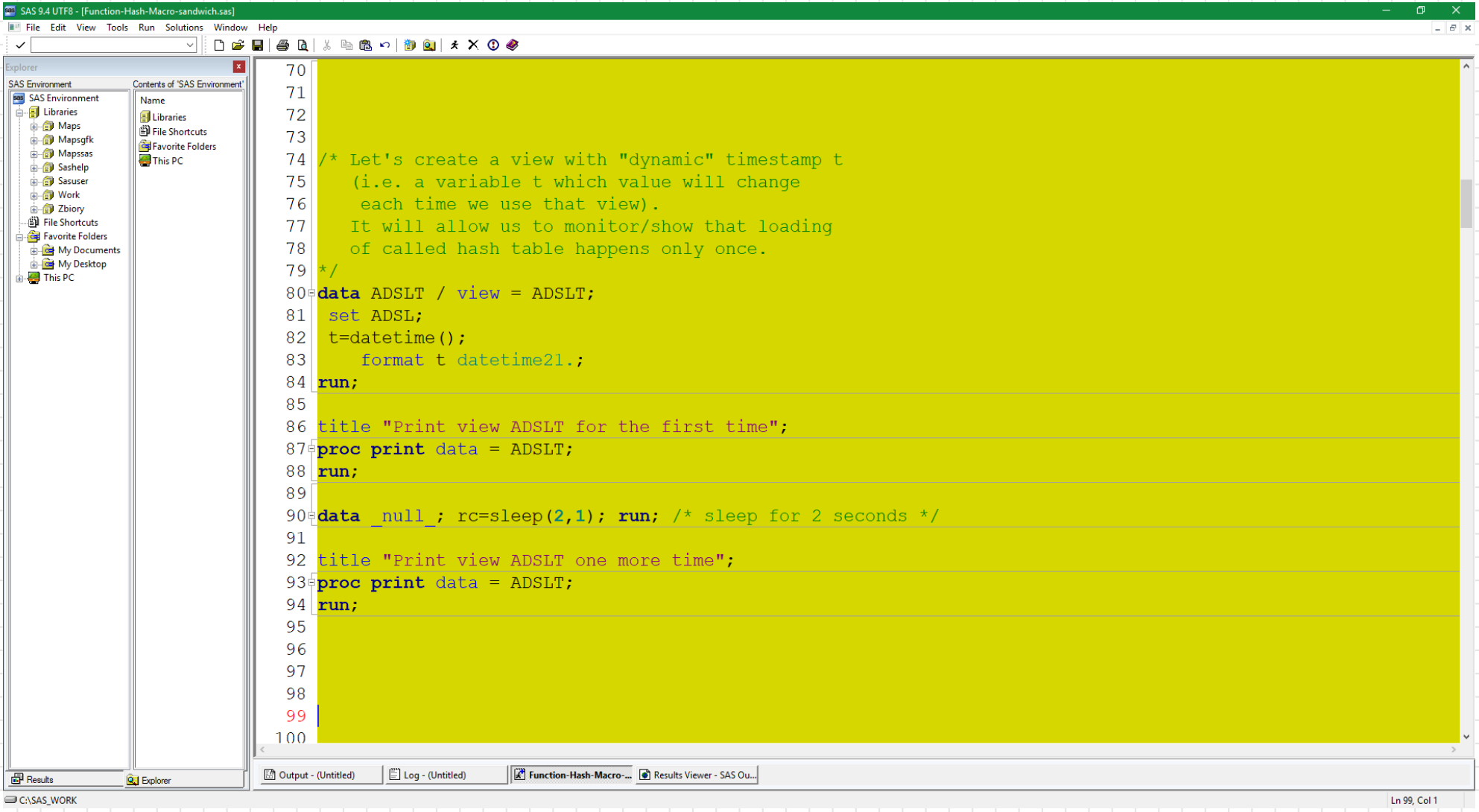
Page: `http://www.mini.pw.edu.pl/~bjablons/SASpublic/`

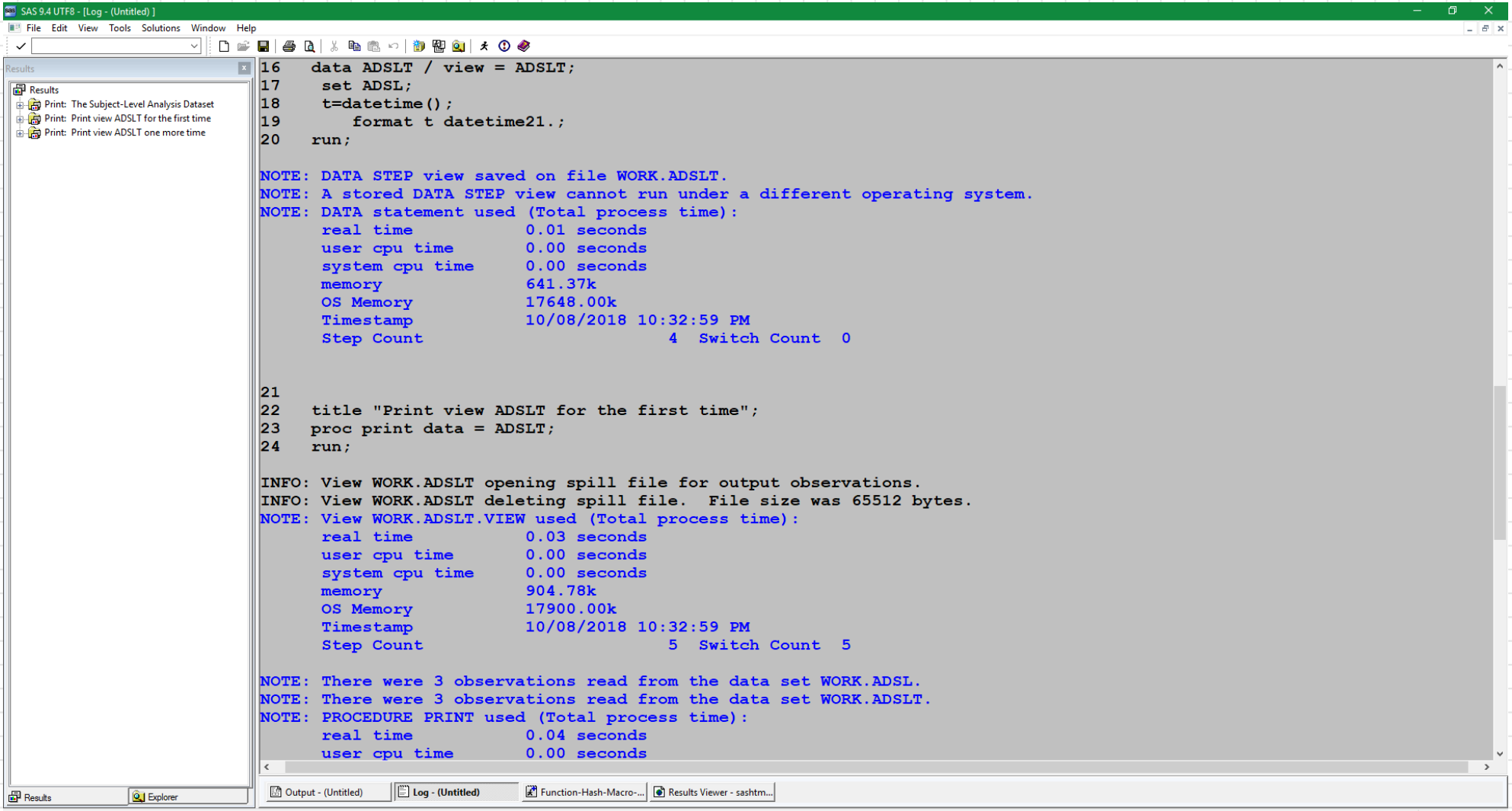
Code: `Function-Hash-Macro-sandwich.sas`
(supplement
`Function-Hash-Macro-sandwich-approach-2.sas`)

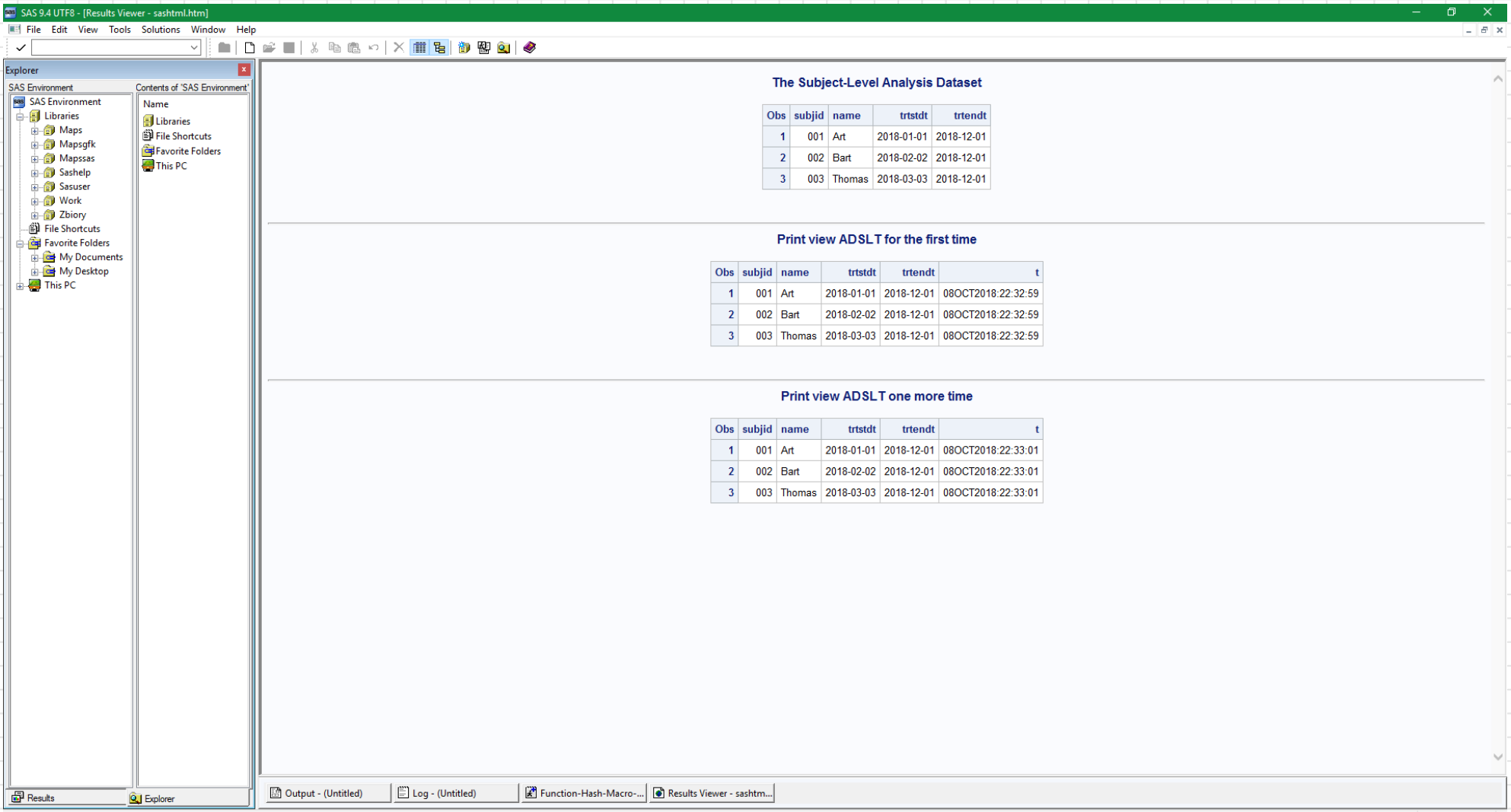

SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas]
File Edit View Tools Run Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
43
44
45 /* The goal is to create a function which allows us
46 to use hash table look-ups in datasteps but without
47 re-loading hash table in every single datastep.
48 */
49
50
51
52
53 /* Let's assume that we have a following almost "Subject-Level Analysis Dataset".
54 */
55 data ADSL;
56 input @1 subjid @3 name : $ 6. @10 trtstdt date9. @20 trtendt date9.;
57 format subjid z3. trtstdt trtendt yymmdd10.;
58 cards;
59 1 Art 1jan2018 31dec2018
60 2 Bart 2feb2018 31dec2018
61 3 Thomas 3mar2018 31dec2018
62 ;
63 run;
64
65 ods html;
66 ods listing close;
67 title "The Subject-Level Analysis Dataset";
68 proc print data = adsl;
69 run;
70
71
72
73
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
C:\SAS_WORK
Ln 45, Col 52

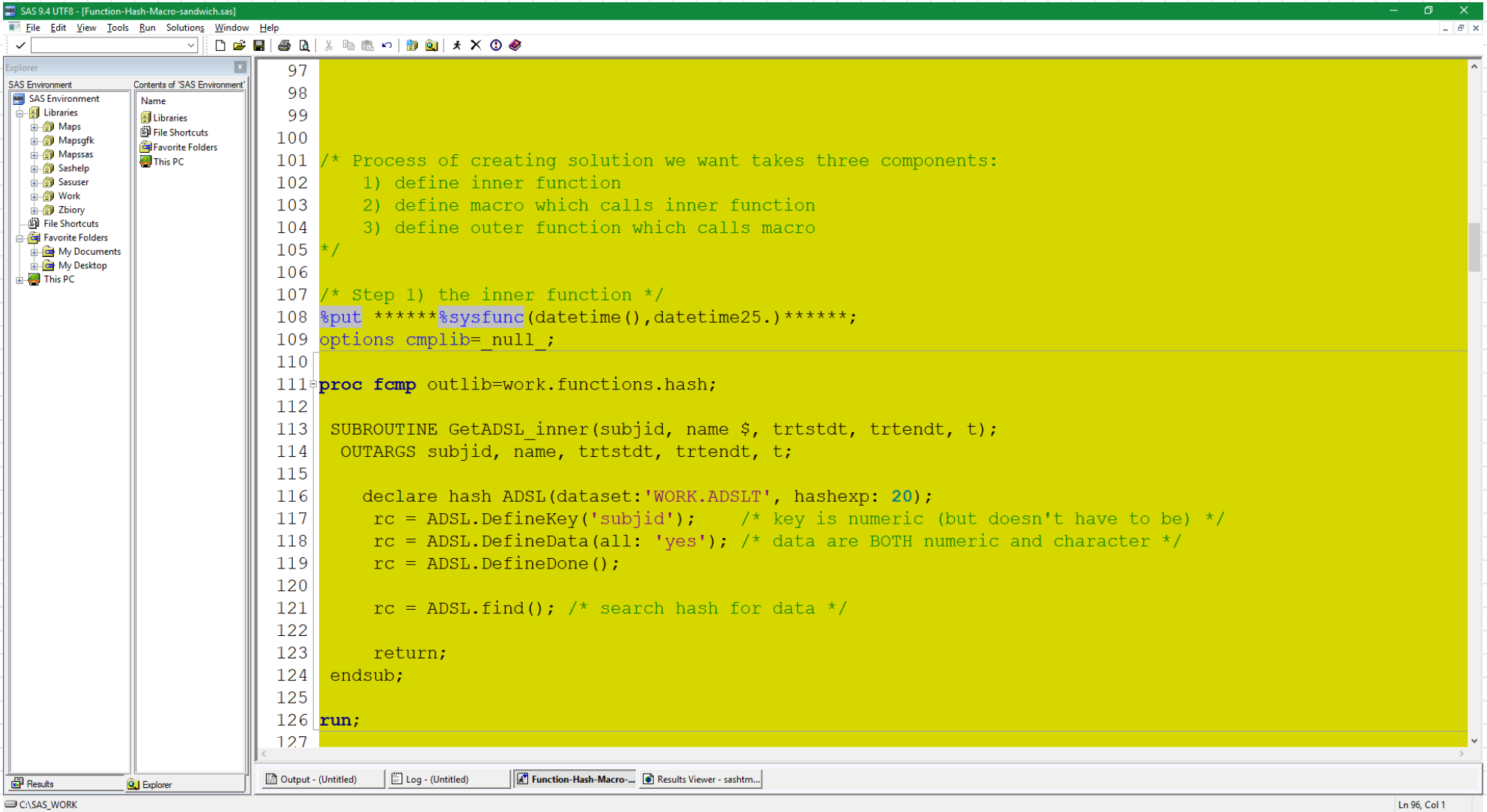

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
1 data ADSL;
2 input @1 subjid @3 name : $ 6. @10 trtstdt date9. @20 trtendt date9.;
3 format subjid z3. trtstdt trtendt yymmdd10.;
4 cards;
NOTE: The data set WORK.ADSL has 3 observations and 4 variables.
NOTE: DATA statement used (Total process time):
real time 0.03 seconds
user cpu time 0.00 seconds
system cpu time 0.01 seconds
memory 407.06k
OS Memory 12008.00k
Timestamp 10/08/2018 10:29:40 PM
Step Count 2 Switch Count 0
8 ;
9 run;
10
11 ods html;
NOTE: Writing HTML Body file: sashtml.htm
12 ods listing close;
13 title "The Subject-Level Analysis Dataset";
14 proc print data = adsl;
15 run;
NOTE: There were 3 observations read from the data set WORK.ADSL.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.03 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 1356.96k
OS Memory 17388.00k
Timestamp 10/08/2018 10:29:40 PM
Step Count 3 Switch Count 0
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - SAS Ou...
C:\SAS_WORK

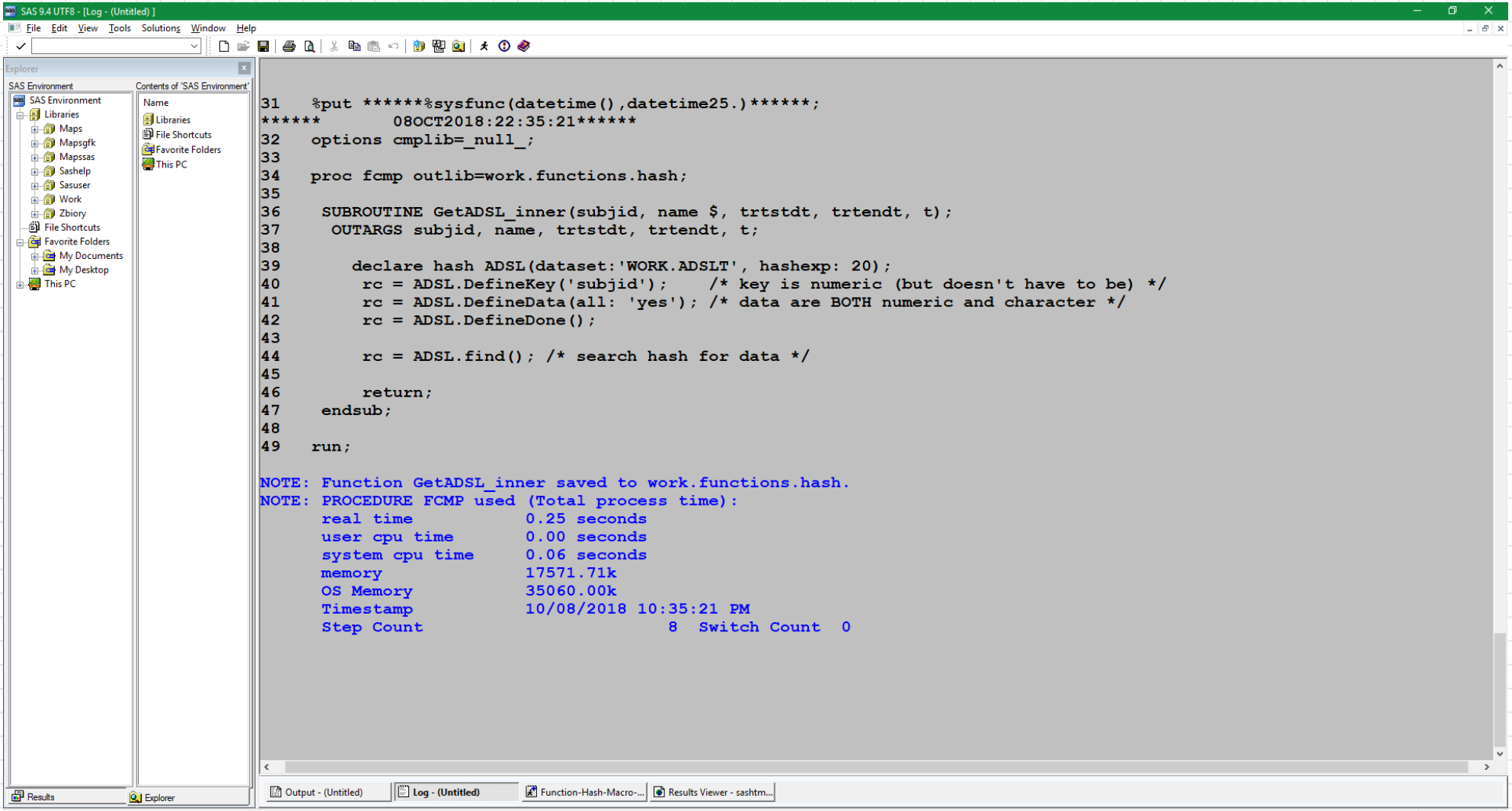

SAS 9.4 UTF8 - [Results Viewer - SAS Output]
File Edit View Go Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
The Subject-Level Analysis Dataset
Obs subjid name trtstdt trtendt
1 001 Art 2018-01-01 2018-12-01
2 002 Bart 2018-02-02 2018-12-01
3 003 Thomas 2018-03-03 2018-12-01
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - SAS ...
C:\SAS_WORK

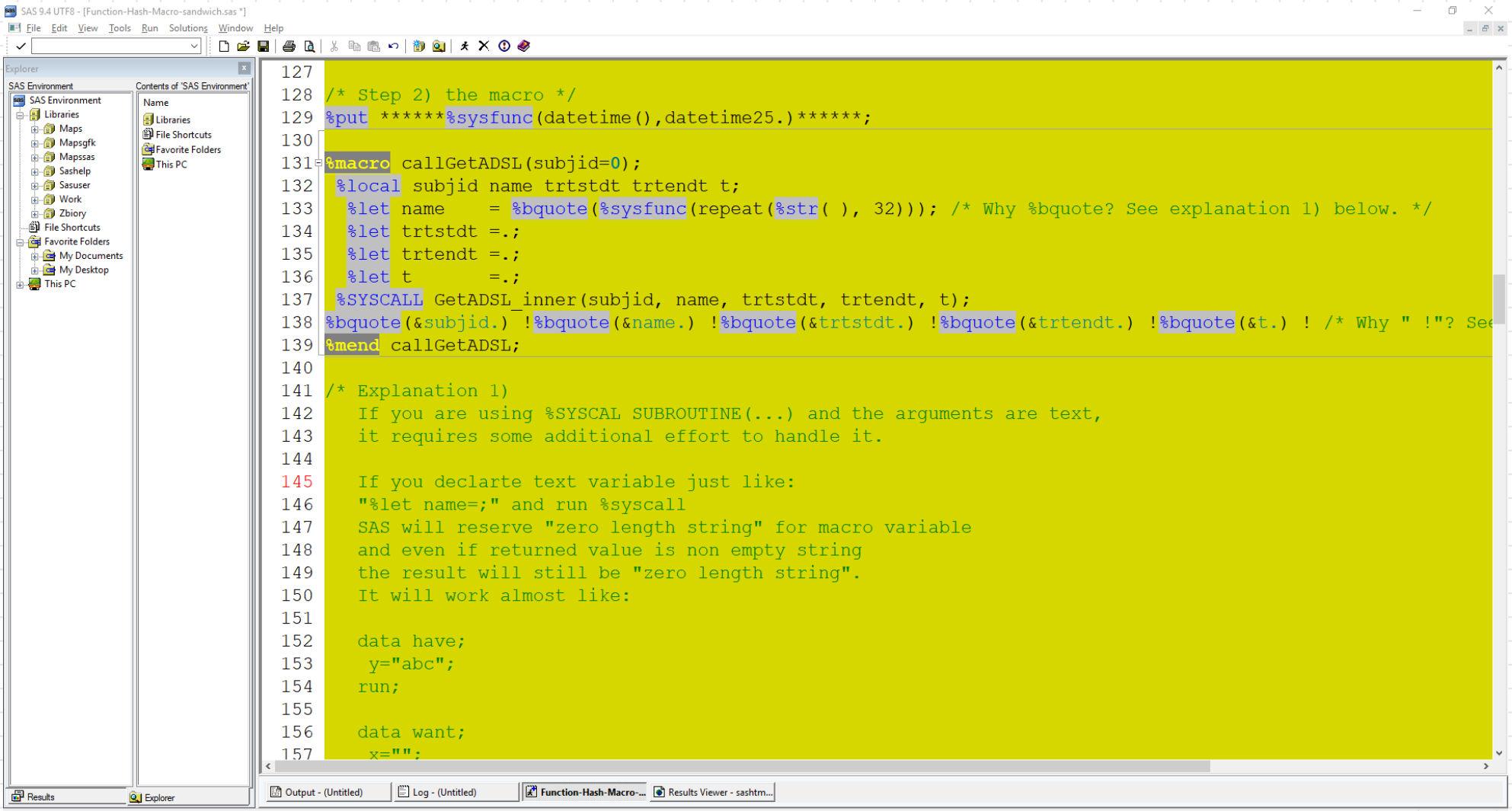

SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas]
File Edit View Tools Run Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
70
71
72
73
74 /* Let's create a view with "dynamic" timestamp t
75 (i.e. a variable t which value will change
76 each time we use that view).
77 It will allow us to monitor/show that loading
78 of called hash table happens only once.
79 */
80 data ADSLT / view = ADSLT;
81 set ADSL;
82 t=datetime();
83 format t datetime21.;
84 run;
85
86 title "Print view ADSLT for the first time";
87 proc print data = ADSLT;
88 run;
89
90 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
91
92 title "Print view ADSLT one more time";
93 proc print data = ADSLT;
94 run;
95
96
97
98
99
100
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - SAS Ou...
C:\SAS_WORK
Ln 99, Col 1

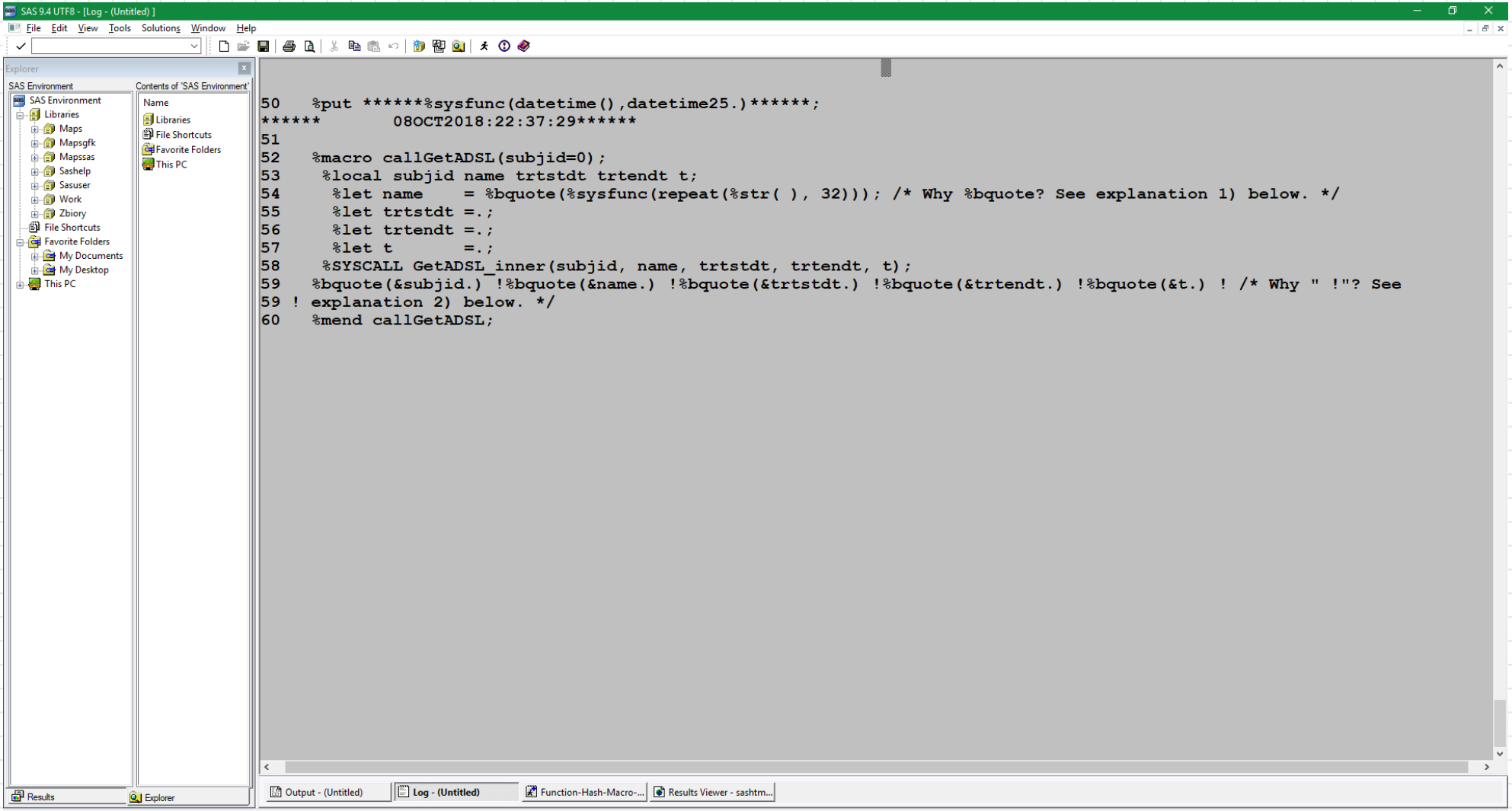

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
16 data ADSLT / view = ADSLT;
17 set ADSL;
18 t=datetime();
19 format t datetime21.;
20 run;
NOTE: DATA STEP view saved on file WORK.ADSLT.
NOTE: A stored DATA STEP view cannot run under a different operating system.
NOTE: DATA statement used (Total process time):
real time 0.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 641.37k
OS Memory 17648.00k
Timestamp 10/08/2018 10:32:59 PM
Step Count 4 Switch Count 0
21
22 title "Print view ADSLT for the first time";
23 proc print data = ADSLT;
24 run;
INFO: View WORK.ADSLT opening spill file for output observations.
INFO: View WORK.ADSLT deleting spill file. File size was 65512 bytes.
NOTE: View WORK.ADSLT.VIEW used (Total process time):
real time 0.03 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 904.78k
OS Memory 17900.00k
Timestamp 10/08/2018 10:32:59 PM
Step Count 5 Switch Count 5
NOTE: There were 3 observations read from the data set WORK.ADSL.
NOTE: There were 3 observations read from the data set WORK.ADSLT.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.04 seconds
user cpu time 0.00 seconds
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

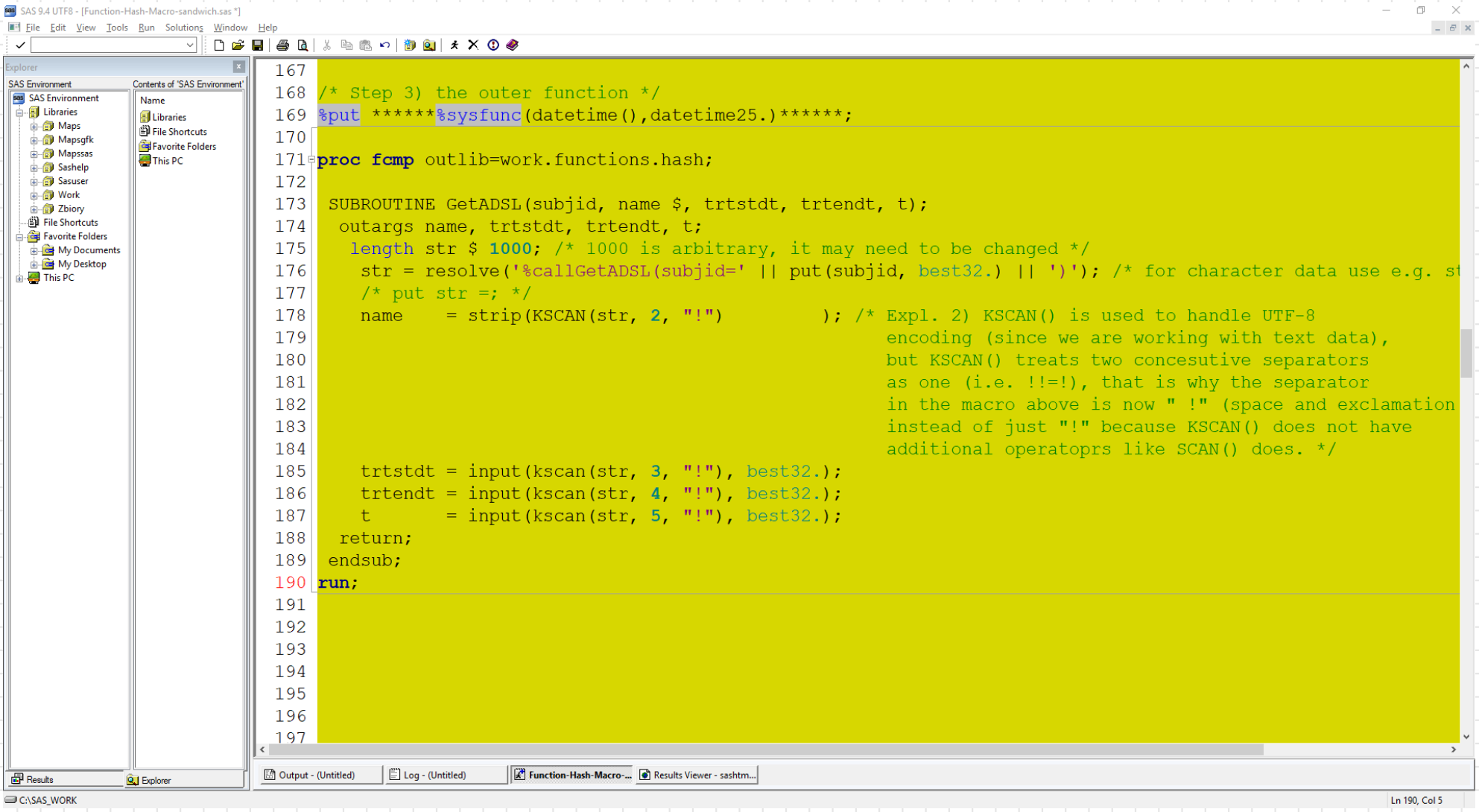

SAS 9.4 UTF8 - [Results Viewer - sashtml.htm]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
Results
Explorer
The Subject-Level Analysis Dataset
Obs subjid name trtstdt trtendt
1 001 Art 2018-01-01 2018-12-01
2 002 Bart 2018-02-02 2018-12-01
3 003 Thomas 2018-03-03 2018-12-01
Print view ADSLT for the first time
Obs subjid name trtstdt trtendt t
1 001 Art 2018-01-01 2018-12-01 08OCT2018:22:32:59
2 002 Bart 2018-02-02 2018-12-01 08OCT2018:22:32:59
3 003 Thomas 2018-03-03 2018-12-01 08OCT2018:22:32:59
Print view ADSLT one more time
Obs subjid name trtstdt trtendt t
1 001 Art 2018-01-01 2018-12-01 08OCT2018:22:33:01
2 002 Bart 2018-02-02 2018-12-01 08OCT2018:22:33:01
3 003 Thomas 2018-03-03 2018-12-01 08OCT2018:22:33:01
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas]
File Edit View Tools Run Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
97
98
99
100
101 /* Process of creating solution we want takes three components:
102 1) define inner function
103 2) define macro which calls inner function
104 3) define outer function which calls macro
105 */
106
107 /* Step 1) the inner function */
108 %put ******%sysfunc(datetime(),datetime25.)******;
109 options cmplib=_null_;
110
111 proc fcmp outlib=work.functions.hash;
112
113 SUBROUTINE GetADSL_inner(subjid, name $, trtstdt, trtendt, t);
114 OUTARGS subjid, name, trtstdt, trtendt, t;
115
116 declare hash ADSL(dataset:'WORK.ADSLT', hashexp: 20);
117 rc = ADSL.DefineKey('subjid'); /* key is numeric (but doesn't have to be) */
118 rc = ADSL.DefineData(all: 'yes'); /* data are BOTH numeric and character */
119 rc = ADSL.DefineDone();
120
121 rc = ADSL.find(); /* search hash for data */
122
123 return;
124 endsub;
125
126 run;
127
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK
Ln 96, Col 1

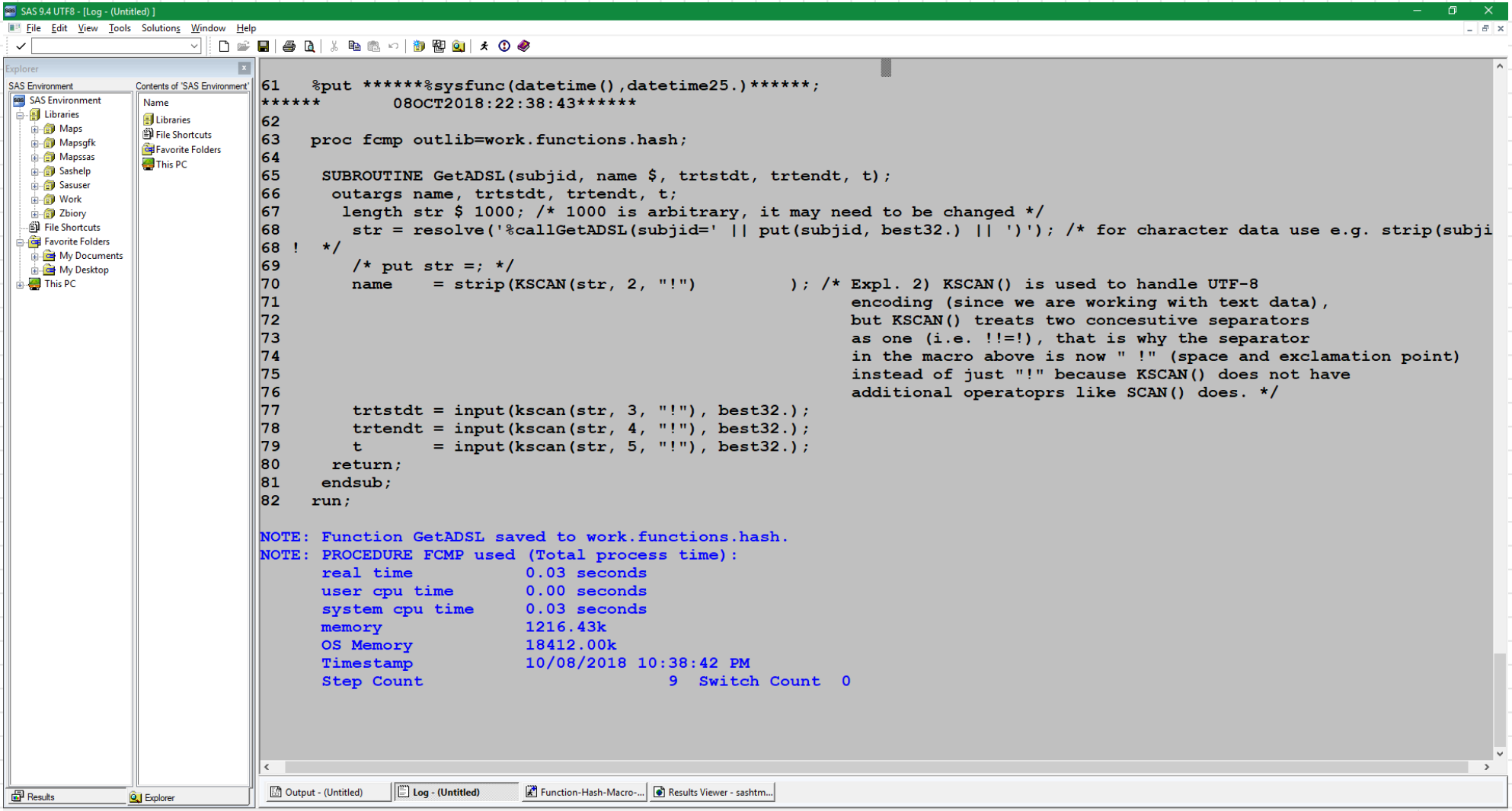

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
31 %put ******%sysfunc(datetime(),datetime25.)******;
****** 08OCT2018:22:35:21******
32 options cmplib=_null_;
33
34 proc fcmp outlib=work.functions.hash;
35
36 SUBROUTINE GetADSL_inner(subjid, name $, trtstdt, trtendt, t);
37 OUTARGS subjid, name, trtstdt, trtendt, t;
38
39 declare hash ADSL(dataset:'WORK.ADSLT', hashexp: 20);
40 rc = ADSL.DefineKey('subjid'); /* key is numeric (but doesn't have to be) */
41 rc = ADSL.DefineData(all: 'yes'); /* data are BOTH numeric and character */
42 rc = ADSL.DefineDone();
43
44 rc = ADSL.find(); /* search hash for data */
45
46 return;
47 endsub;
48
49 run;
NOTE: Function GetADSL_inner saved to work.functions.hash.
NOTE: PROCEDURE FCMP used (Total process time):
real time 0.25 seconds
user cpu time 0.00 seconds
system cpu time 0.06 seconds
memory 17571.71k
OS Memory 35060.00k
Timestamp 10/08/2018 10:35:21 PM
Step Count 8 Switch Count 0
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

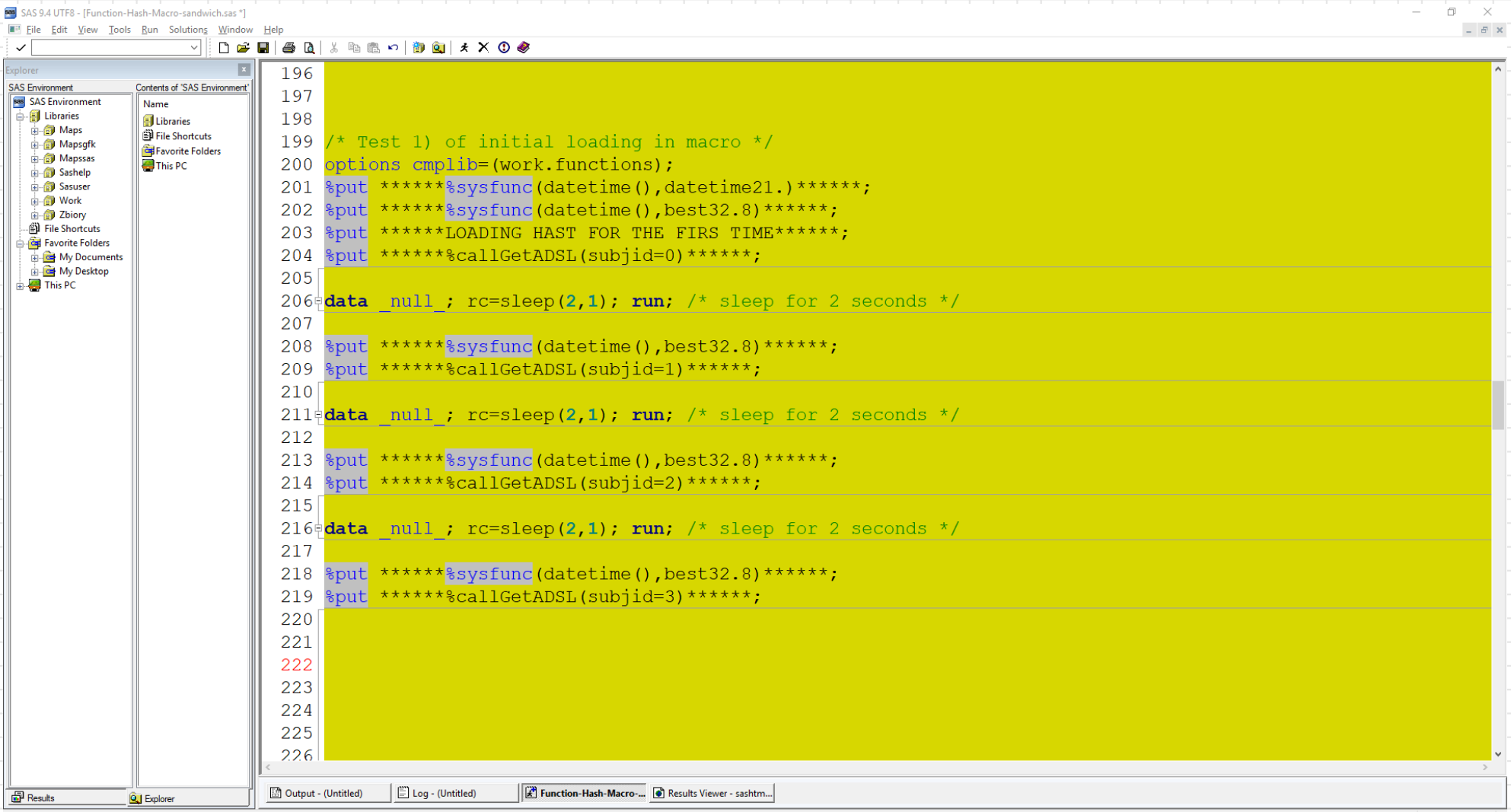

SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas *]
File Edit View Tools Run Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
127
128 /* Step 2) the macro */
129 %put ******%sysfunc(datetime(),datetime25.)******;
130
131 %macro callGetADSL(subjid=0);
132 %local subjid name trtstdt trtendt t;
133 %let name = %bquote(%sysfunc(repeat(%str( ), 32))); /* Why %bquote? See explanation 1) below. */
134 %let trtstdt =.;
135 %let trtendt =.;
136 %let t =.;
137 %SYSCALL GetADSL_inner(subjid, name, trtstdt, trtendt, t);
138 %bquote(&subjid.) !%bquote(&name.) !%bquote(&trtstdt.) !%bquote(&trtendt.) !%bquote(&t.) ! /* Why " !"? See
139 %mend callGetADSL;
140
141 /* Explanation 1)
142 If you are using %SYSCAL SUBROUTINE(...) and the arguments are text,
143 it requires some additional effort to handle it.
144
145 If you declarte text variable just like:
146 "%let name=;" and run %syscall
147 SAS will reserve "zero length string" for macro variable
148 and even if returned value is non empty string
149 the result will still be "zero length string".
150 It will work almost like:
151
152 data have;
153 y="abc";
154 run;
155
156 data want;
157 x="";
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK
Ln 145, Col 44

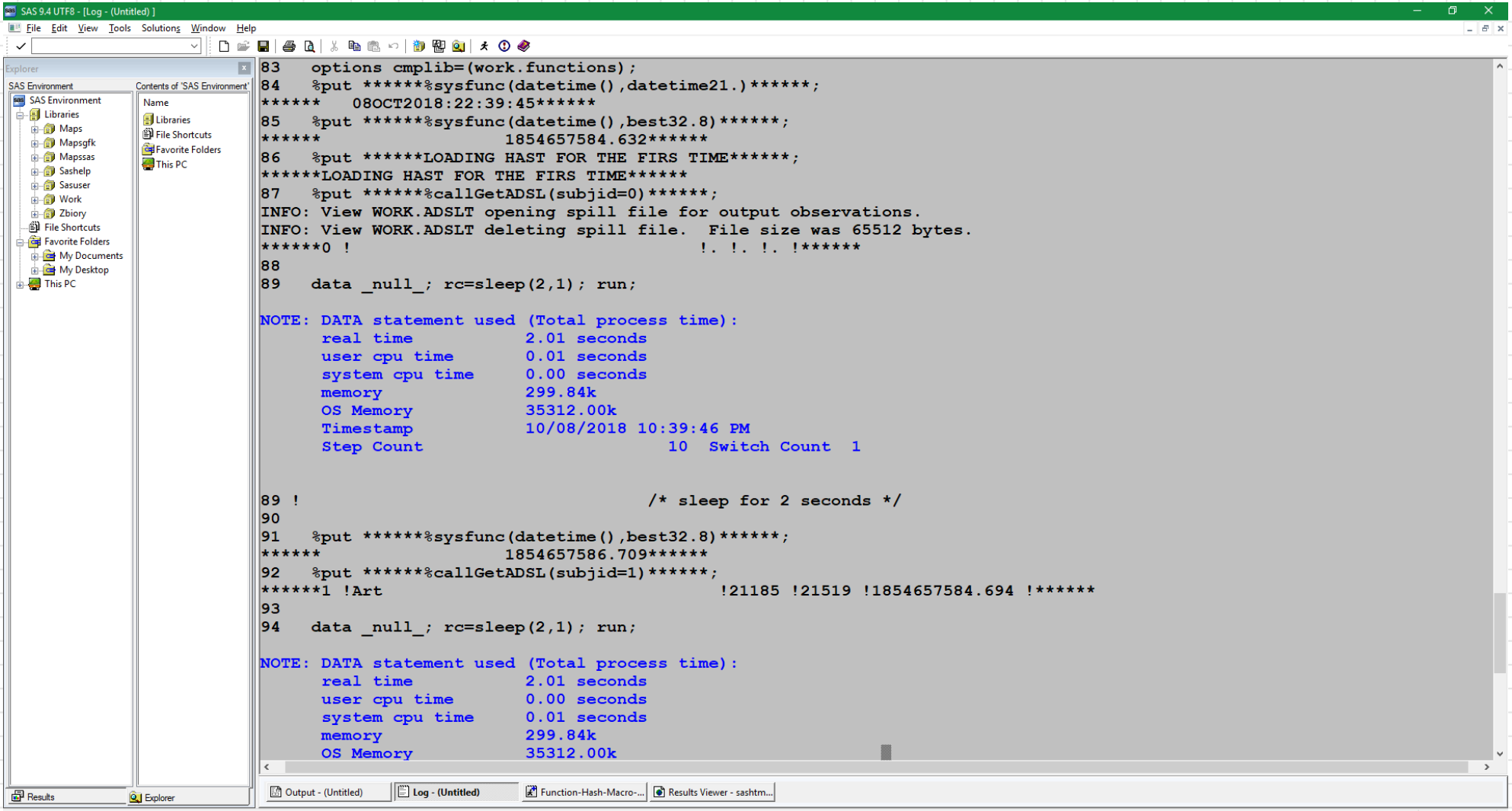

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
50 %put ******%sysfunc(datetime(),datetime25.)******;
****** 08OCT2018:22:37:29******
51
52 %macro callGetADSL(subjid=0);
53 %local subjid name trtstdt trtendt t;
54 %let name = %bquote(%sysfunc(repeat(%str( ), 32))); /* Why %bquote? See explanation 1) below. */
55 %let trtstdt =.;
56 %let trtendt =.;
57 %let t =.;
58 %SYSCALL GetADSL_inner(subjid, name, trtstdt, trtendt, t);
59 %bquote(&subjid.) !%bquote(&name.) !%bquote(&trtstdt.) !%bquote(&trtendt.) !%bquote(&t.) ! /* Why " !"? See
59 ! explanation 2) below. */
60 %mend callGetADSL;
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

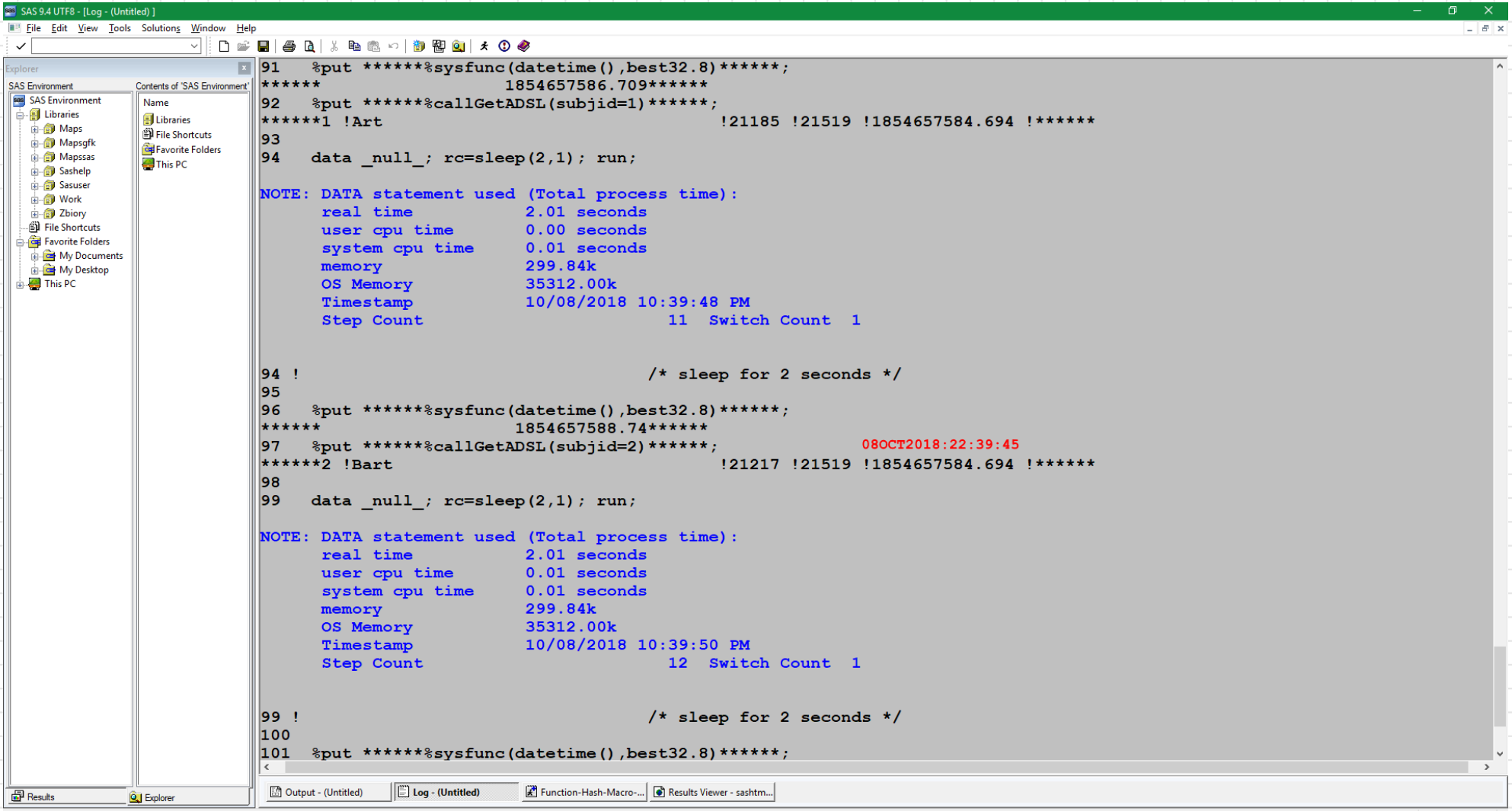


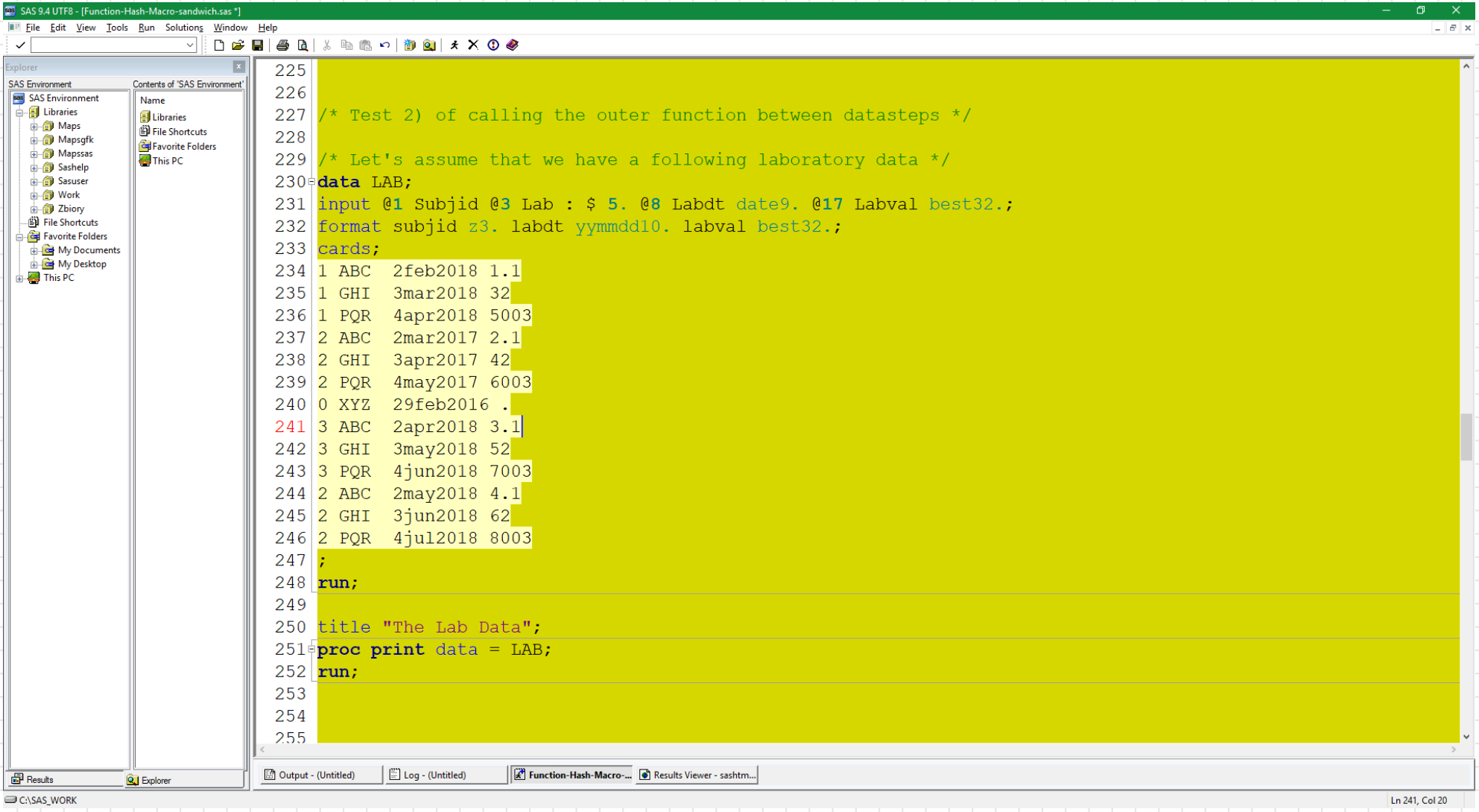

```
167
168 /* Step 3) the outer function */
169 %put ******%sysfunc(datetime(),datetime25.)******;
170
171 proc fcmp outlib=work.functions.hash;
172
173  SUBROUTINE GetADSL(subjid, name $, trtstdt, trtendt, t);
174   outargs name, trtstdt, trtendt, t;
175    length str $ 1000; /* 1000 is arbitrary, it may need to be changed */
176     str = resolve('%callGetADSL(subjid=' || put(subjid, best32.) || ')'); /* for character data use e.g. st
177     /* put str =; */
178     name    = strip(KSCAN(str, 2, "!")         ); /* Expl. 2) KSCAN() is used to handle UTF-8
179                                                      encoding (since we are working with text data),
180                                                      but KSCAN() treats two concesutive separators
181                                                      as one (i.e. !!=!), that is why the separator
182                                                      in the macro above is now " !" (space and exclamation
183                                                      instead of just "!" because KSCAN() does not have
184                                                      additional operatoprs like SCAN() does. */
185     trtstdt = input(kscan(str, 3, "!"), best32.);
186     trtendt = input(kscan(str, 4, "!"), best32.);
187     t       = input(kscan(str, 5, "!"), best32.);
188   return;
189  endsub;
190 run;
191
192
193
194
195
196
197
```

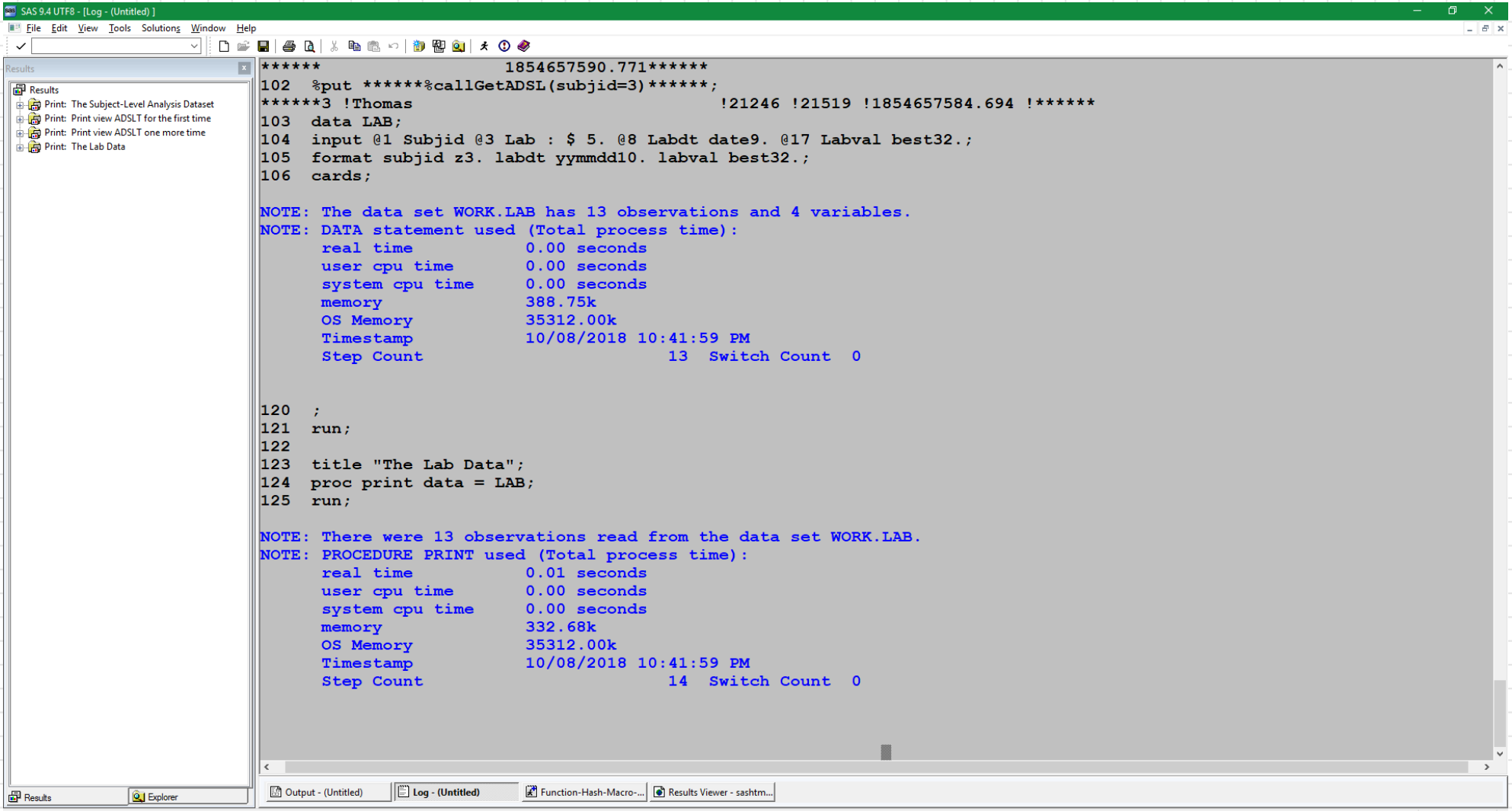

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
61 %put ******%sysfunc(datetime(),datetime25.)******;
****** 08OCT2018:22:38:43******
62
63 proc fcmp outlib=work.functions.hash;
64
65 SUBROUTINE GetADSL(subjid, name $, trtstdt, trtendt, t);
66 outargs name, trtstdt, trtendt, t;
67 length str $ 1000; /* 1000 is arbitrary, it may need to be changed */
68 str = resolve('%callGetADSL(subjid=' || put(subjid, best32.) || ')'); /* for character data use e.g. strip(subji
68 ! */
69 /* put str =; */
70 name = strip(KSCAN(str, 2, "!")); /* Expl. 2) KSCAN() is used to handle UTF-8
71 encoding (since we are working with text data),
72 but KSCAN() treats two concesutive separators
73 as one (i.e. !!=!), that is why the separator
74 in the macro above is now " !" (space and exclamation point)
75 instead of just "!" because KSCAN() does not have
76 additional operatoprs like SCAN() does. */
77 trtstdt = input(kscan(str, 3, "!"), best32.);
78 trtendt = input(kscan(str, 4, "!"), best32.);
79 t = input(kscan(str, 5, "!"), best32.);
80 return;
81 endsub;
82 run;
NOTE: Function GetADSL saved to work.functions.hash.
NOTE: PROCEDURE FCMP used (Total process time):
real time 0.03 seconds
user cpu time 0.00 seconds
system cpu time 0.03 seconds
memory 1216.43k
OS Memory 18412.00k
Timestamp 10/08/2018 10:38:42 PM
Step Count 9 Switch Count 0
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas *]
File Edit View Tools Run Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
196
197
198
199 /* Test 1) of initial loading in macro */
200 options cmplib=(work.functions);
201 %put ******%sysfunc(datetime(),datetime21.)******;
202 %put ******%sysfunc(datetime(),best32.8)******;
203 %put ******LOADING HAST FOR THE FIRS TIME******;
204 %put ******%callGetADSL(subjid=0)******;
205
206 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
207
208 %put ******%sysfunc(datetime(),best32.8)******;
209 %put ******%callGetADSL(subjid=1)******;
210
211 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
212
213 %put ******%sysfunc(datetime(),best32.8)******;
214 %put ******%callGetADSL(subjid=2)******;
215
216 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
217
218 %put ******%sysfunc(datetime(),best32.8)******;
219 %put ******%callGetADSL(subjid=3)******;
220
221
222
223
224
225
226
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK
Ln 222, Col 1


SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
Results
Explorer
83 options cmplib=(work.functions);
84 %put ******%sysfunc(datetime(),datetime21.)******;
****** 08OCT2018:22:39:45******
85 %put ******%sysfunc(datetime(),best32.8)******;
****** 1854657584.632******
86 %put ******LOADING HAST FOR THE FIRS TIME******;
******LOADING HAST FOR THE FIRS TIME******
87 %put ******%callGetADSL(subjid=0)******;
INFO: View WORK.ADSLT opening spill file for output observations.
INFO: View WORK.ADSLT deleting spill file. File size was 65512 bytes.
******0 ! !. !. !. !******
88
89 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.01 seconds
system cpu time 0.00 seconds
memory 299.84k
OS Memory 35312.00k
Timestamp 10/08/2018 10:39:46 PM
Step Count 10 Switch Count 1
89 ! /* sleep for 2 seconds */
90
91 %put ******%sysfunc(datetime(),best32.8)******;
****** 1854657586.709******
92 %put ******%callGetADSL(subjid=1)******;
******1 !Art !21185 !21519 !1854657584.694 !******
93
94 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.00 seconds
system cpu time 0.01 seconds
memory 299.84k
OS Memory 35312.00k
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
91 %put ******%sysfunc(datetime(),best32.8)******;
****** 1854657586.709******
92 %put ******%callGetADSL(subjid=1)******;
******1 !Art !21185 !21519 !1854657584.694 !******
93
94 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.00 seconds
system cpu time 0.01 seconds
memory 299.84k
OS Memory 35312.00k
Timestamp 10/08/2018 10:39:48 PM
Step Count 11 Switch Count 1
94 ! /* sleep for 2 seconds */
95
96 %put ******%sysfunc(datetime(),best32.8)******;
****** 1854657588.74******
97 %put ******%callGetADSL(subjid=2)******;
08OCT2018:22:39:45
******2 !Bart !21217 !21519 !1854657584.694 !******
98
99 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.01 seconds
system cpu time 0.01 seconds
memory 299.84k
OS Memory 35312.00k
Timestamp 10/08/2018 10:39:50 PM
Step Count 12 Switch Count 1
99 ! /* sleep for 2 seconds */
100
101 %put ******%sysfunc(datetime(),best32.8)******;
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas *]
File Edit View Tools Run Solutions Window Help
Explorer
SAS Environment
Contents of 'SAS Environment'
SAS Environment
Libraries
Maps
Mapsgfk
Mapssas
Sashelp
Sasuser
Work
Zbiory
File Shortcuts
Favorite Folders
My Documents
My Desktop
This PC
Name
Libraries
File Shortcuts
Favorite Folders
This PC
225
226
227 /* Test 2) of calling the outer function between datasteps */
228
229 /* Let's assume that we have a following laboratory data */
230 data LAB;
231 input @1 Subjid @3 Lab : $ 5. @8 Labdt date9. @17 Labval best32.;
232 format subjid z3. labdt yymmdd10. labval best32.;
233 cards;
234 1 ABC 2feb2018 1.1
235 1 GHI 3mar2018 32
236 1 PQR 4apr2018 5003
237 2 ABC 2mar2017 2.1
238 2 GHI 3apr2017 42
239 2 PQR 4may2017 6003
240 0 XYZ 29feb2016 .
241 3 ABC 2apr2018 3.1
242 3 GHI 3may2018 52
243 3 PQR 4jun2018 7003
244 2 ABC 2may2018 4.1
245 2 GHI 3jun2018 62
246 2 PQR 4jul2018 8003
247 ;
248 run;
249
250 title "The Lab Data";
251 proc print data = LAB;
252 run;
253
254
255
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK
Ln 241, Col 20


SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
****** 1854657590.771******
102 %put ******%callGetADSL(subjid=3)******;
******3 !Thomas !21246 !21519 !1854657584.694 !******
103 data LAB;
104 input @1 Subjid @3 Lab : $ 5. @8 Labdt date9. @17 Labval best32.;
105 format subjid z3. labdt yymmdd10. labval best32.;
106 cards;
NOTE: The data set WORK.LAB has 13 observations and 4 variables.
NOTE: DATA statement used (Total process time):
real time 0.00 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 388.75k
OS Memory 35312.00k
Timestamp 10/08/2018 10:41:59 PM
Step Count 13 Switch Count 0
120 ;
121 run;
122
123 title "The Lab Data";
124 proc print data = LAB;
125 run;
NOTE: There were 13 observations read from the data set WORK.LAB.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 332.68k
OS Memory 35312.00k
Timestamp 10/08/2018 10:41:59 PM
Step Count 14 Switch Count 0
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

SAS 9.4 UTF8 - [Results Viewer - sashtml.htm]

File Edit View Go Tools Solutions Window Help

Results

- Results
  - Print: The Subject-Level Analysis Dataset
  - Print: Print view ADSLT for the first time
  - Print: Print view ADSLT one more time
  - Print: The Lab Data

Results | Explorer

Print view ADSLT for the first time

Obs	subjid	name	trtstdt	trtendt	t
1	001	Art	2018-01-01	2018-12-01	08OCT2018:22:32:59
2	002	Bart	2018-02-02	2018-12-01	08OCT2018:22:32:59
3	003	Thomas	2018-03-03	2018-12-01	08OCT2018:22:32:59

Print view ADSLT one more time

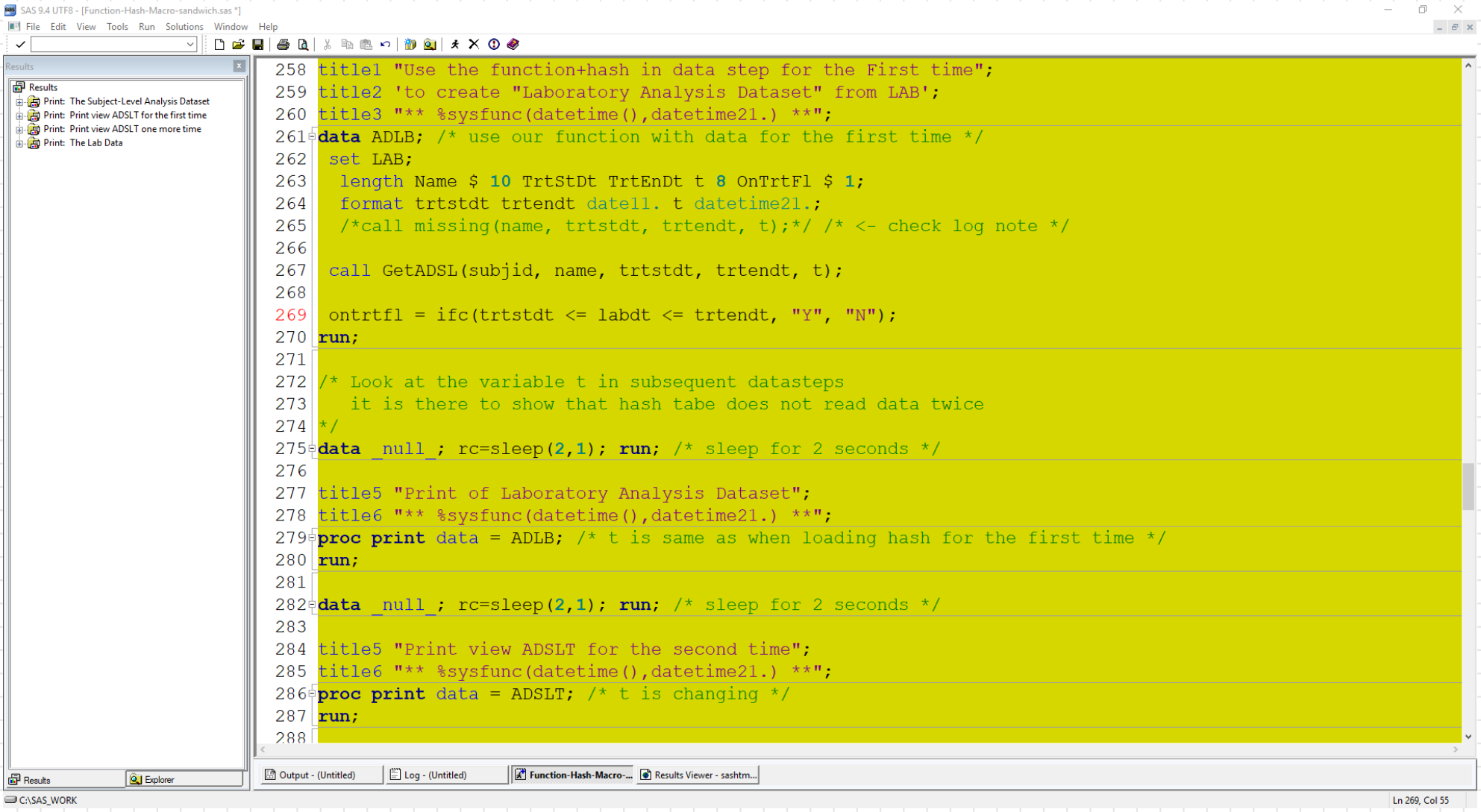
Obs	subjid	name	trtstdt	trtendt	t
1	001	Art	2018-01-01	2018-12-01	08OCT2018:22:33:01
2	002	Bart	2018-02-02	2018-12-01	08OCT2018:22:33:01
3	003	Thomas	2018-03-03	2018-12-01	08OCT2018:22:33:01

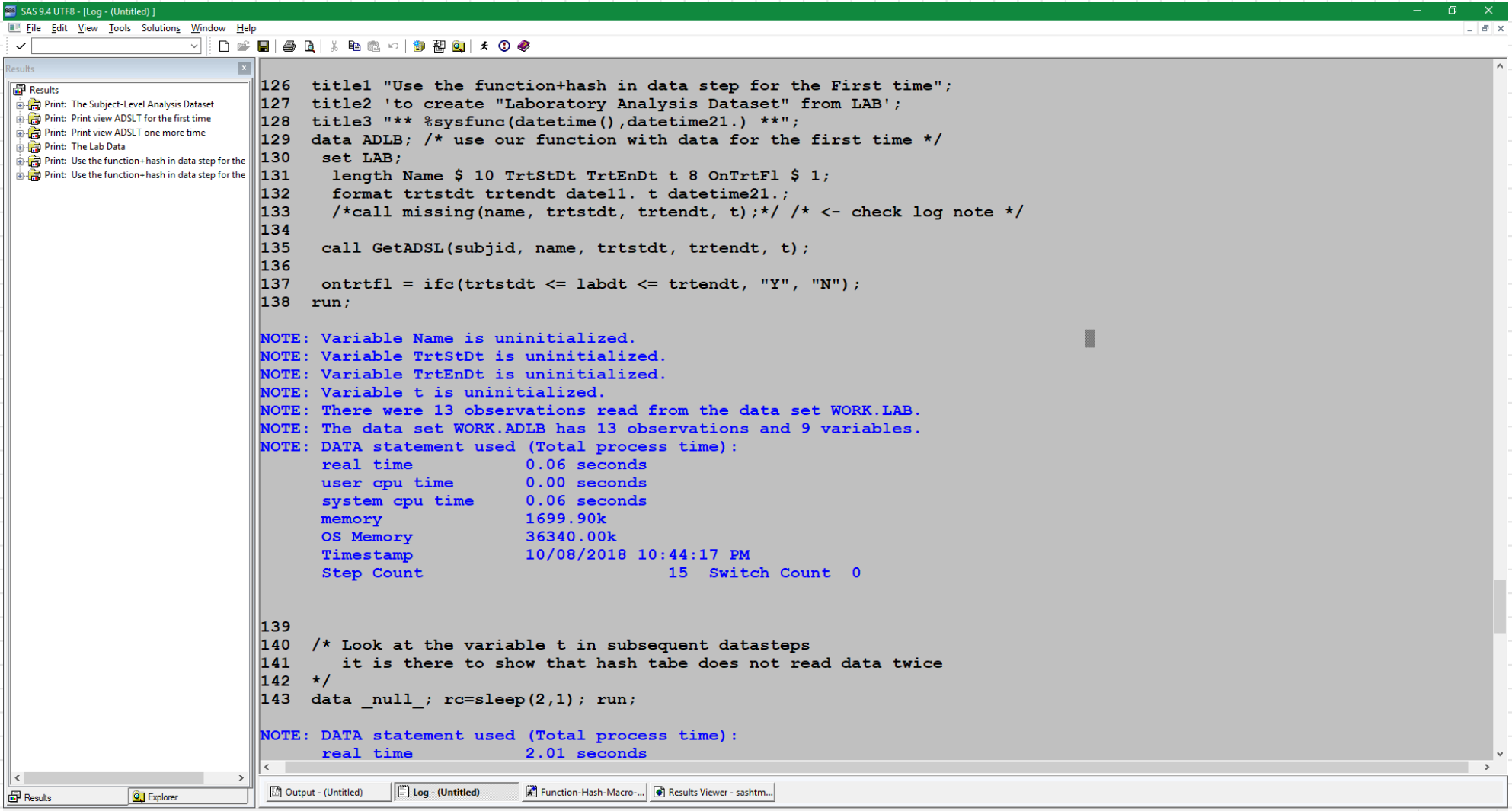
The Lab Data

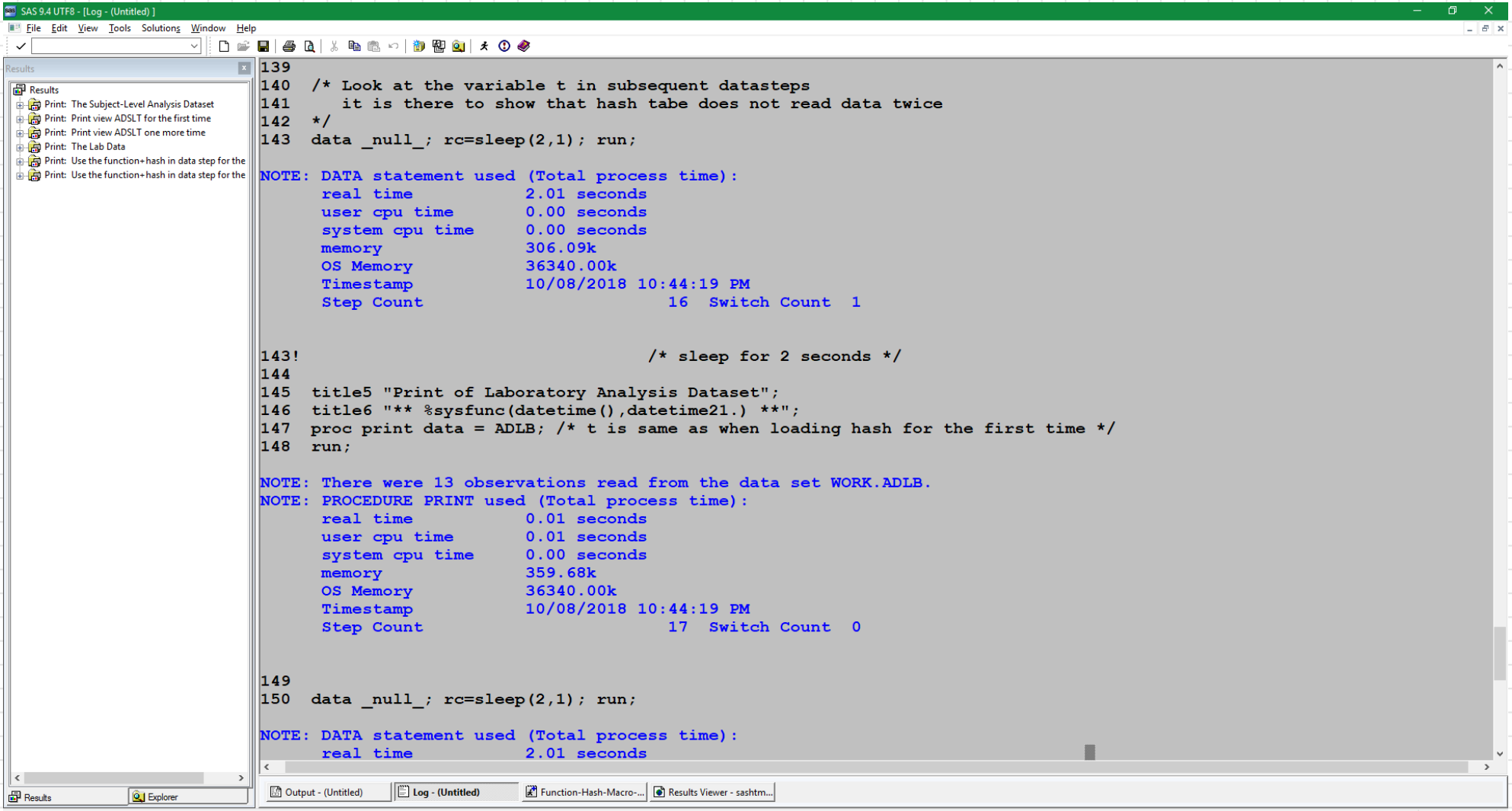
Obs	Subjid	Lab	Labdt	Labval
1	001	ABC	2018-02-02	1.1
2	001	GHI	2018-03-03	32
3	001	PQR	2018-04-04	5003
4	002	ABC	2017-03-02	2.1
5	002	GHI	2017-04-03	42
6	002	PQR	2017-05-04	6003
7	000	XYZ	2016-02-29	.
8	003	ABC	2018-04-02	3.1
9	003	GHI	2018-05-03	52
10	003	PQR	2018-06-04	7003
11	002	ABC	2018-05-02	4.1
12	002	GHI	2018-06-03	62
13	002	PQR	2018-07-04	8003

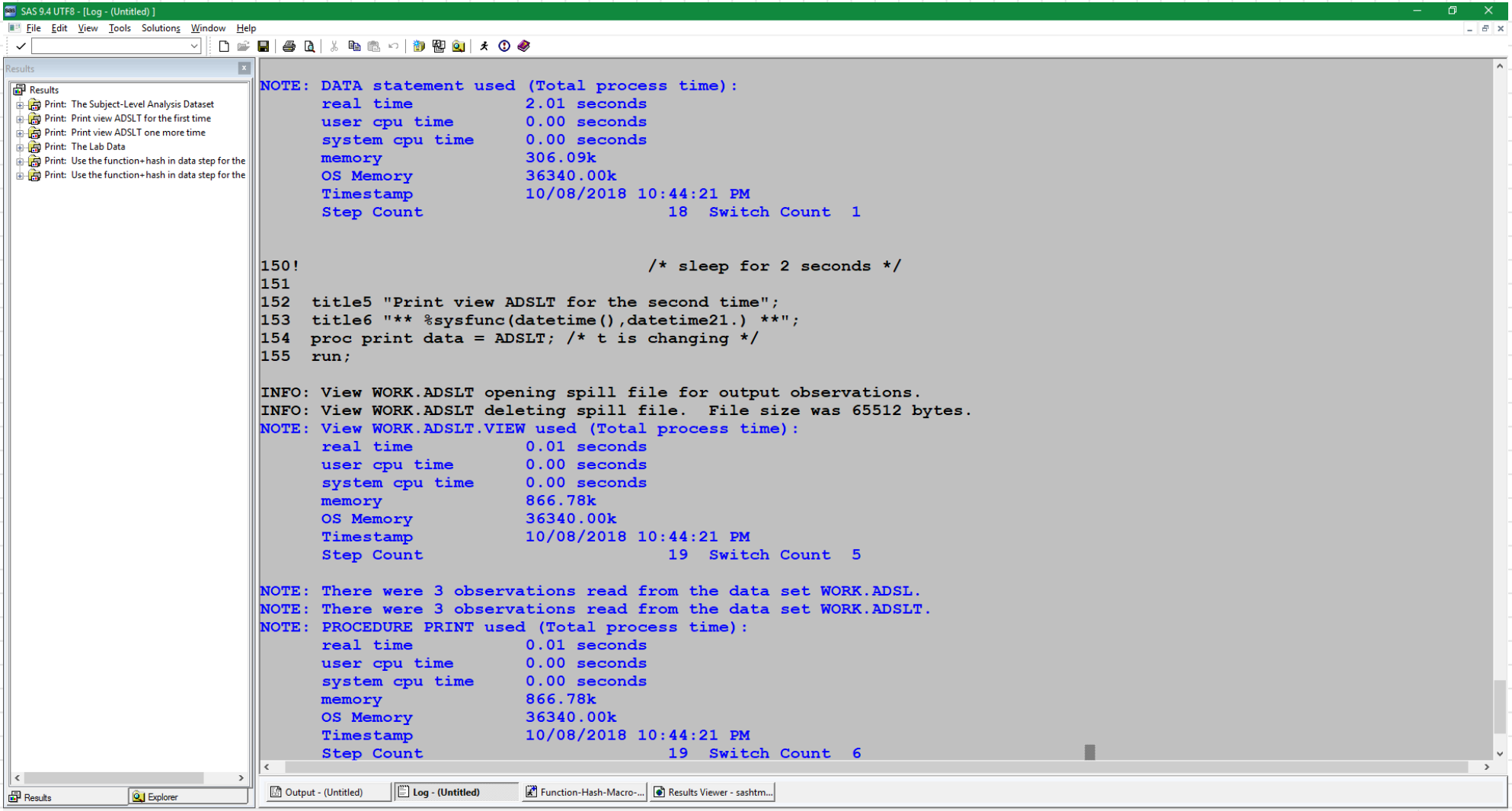
Output - (Untitled) | Log - (Untitled) | Function-Hash-Macro-... | Results Viewer - sasht...

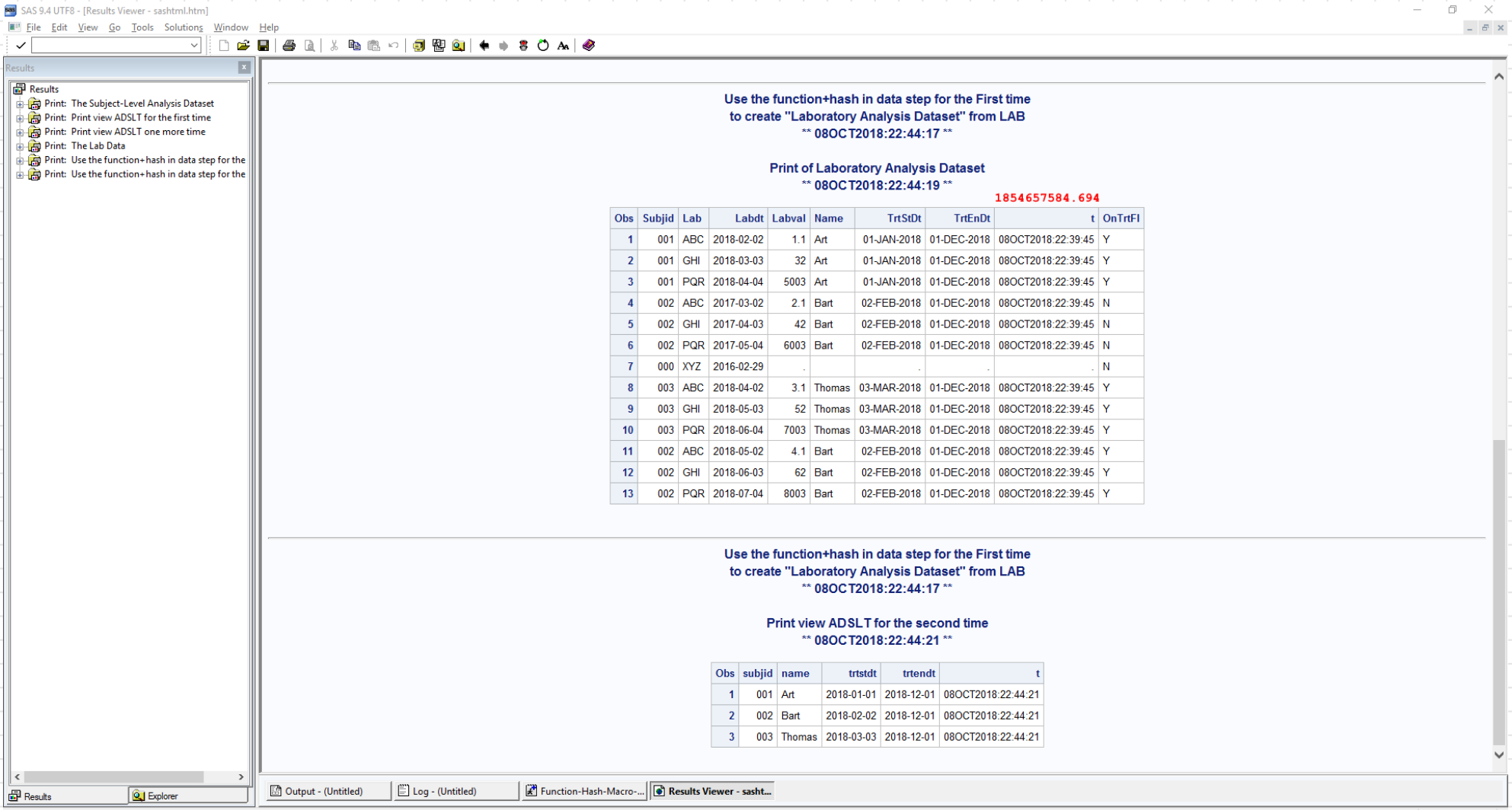
C:\SAS_WORK

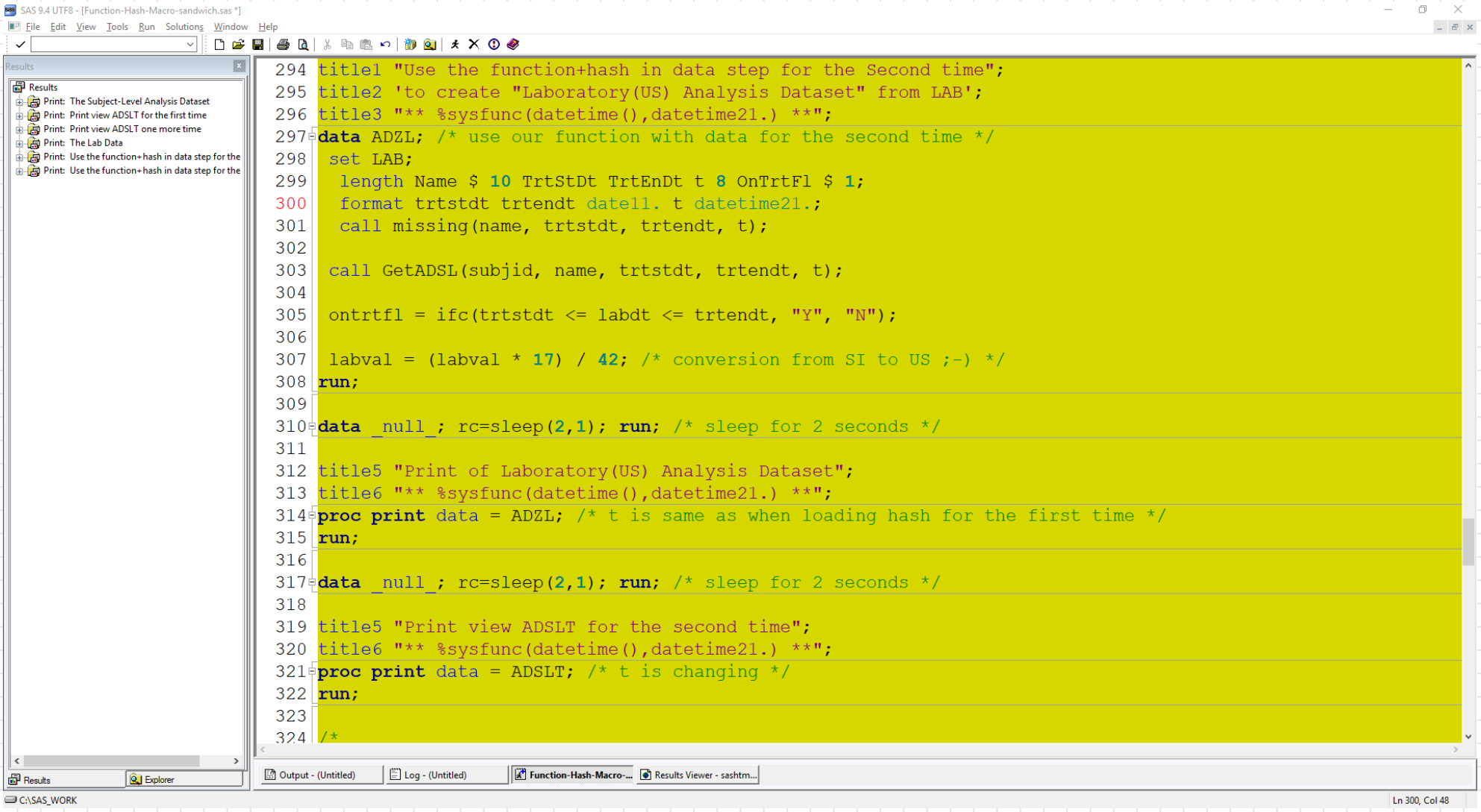

SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas *]
File Edit View Tools Run Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Results
Explorer
258 title1 "Use the function+hash in data step for the First time";
259 title2 'to create "Laboratory Analysis Dataset" from LAB';
260 title3 "** %sysfunc(datetime(),datetime21.) **";
261 data ADLB; /* use our function with data for the first time */
262 set LAB;
263 length Name $ 10 TrtStDt TrtEnDt t 8 OnTrtFl $ 1;
264 format trtstdt trtendt date11. t datetime21.;
265 /*call missing(name, trtstdt, trtendt, t);*/ /* <- check log note */
266
267 call GetADSL(subjid, name, trtstdt, trtendt, t);
268
269 ontrtfl = ifc(trtstdt <= labdt <= trtendt, "Y", "N");
270 run;
271
272 /* Look at the variable t in subsequent datasteps
273 it is there to show that hash tabe does not read data twice
274 */
275 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
276
277 title5 "Print of Laboratory Analysis Dataset";
278 title6 "** %sysfunc(datetime(),datetime21.) **";
279 proc print data = ADLB; /* t is same as when loading hash for the first time */
280 run;
281
282 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
283
284 title5 "Print view ADSLT for the second time";
285 title6 "** %sysfunc(datetime(),datetime21.) **";
286 proc print data = ADSLT; /* t is changing */
287 run;
288
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK
Ln 269, Col 55

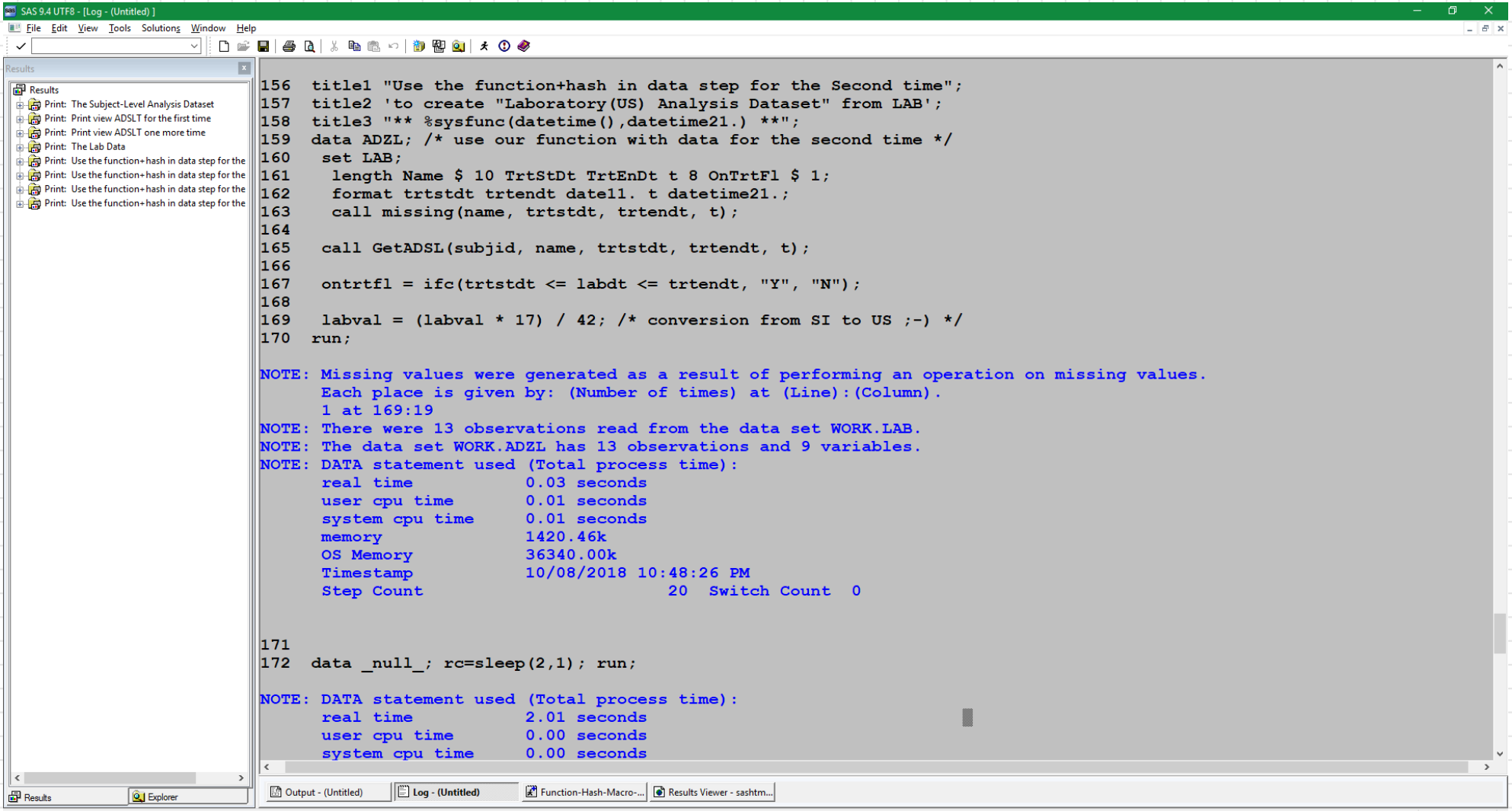

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
126 title1 "Use the function+hash in data step for the First time";
127 title2 'to create "Laboratory Analysis Dataset" from LAB';
128 title3 "** %sysfunc(datetime(),datetime21.) **";
129 data ADLB; /* use our function with data for the first time */
130 set LAB;
131 length Name $ 10 TrtStDt TrtEnDt t 8 OnTrtFl $ 1;
132 format trtstdt trtendt date11. t datetime21.;
133 /*call missing(name, trtstdt, trtendt, t);*/ /* <- check log note */
134
135 call GetADSL(subjid, name, trtstdt, trtendt, t);
136
137 ontrtfl = ifc(trtstdt <= labdt <= trtendt, "Y", "N");
138 run;
NOTE: Variable Name is uninitialized.
NOTE: Variable TrtStDt is uninitialized.
NOTE: Variable TrtEnDt is uninitialized.
NOTE: Variable t is uninitialized.
NOTE: There were 13 observations read from the data set WORK.LAB.
NOTE: The data set WORK.ADLB has 13 observations and 9 variables.
NOTE: DATA statement used (Total process time):
real time 0.06 seconds
user cpu time 0.00 seconds
system cpu time 0.06 seconds
memory 1699.90k
OS Memory 36340.00k
Timestamp 10/08/2018 10:44:17 PM
Step Count 15 Switch Count 0
139
140 /* Look at the variable t in subsequent datasteps
141 it is there to show that hash tabe does not read data twice
142 */
143 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

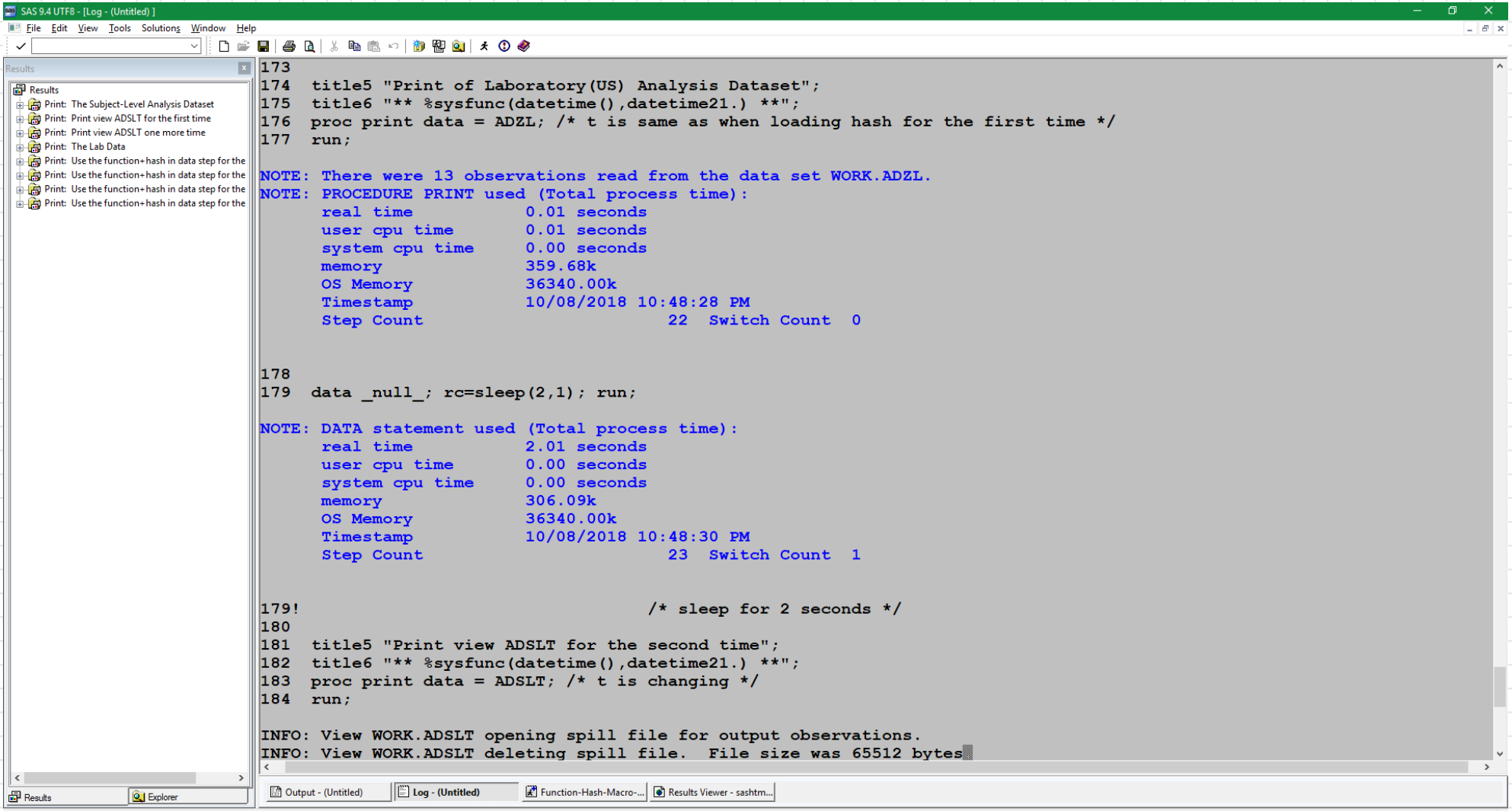

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
139
140 /* Look at the variable t in subsequent datasteps
141 it is there to show that hash tabe does not read data twice
142 */
143 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 306.09k
OS Memory 36340.00k
Timestamp 10/08/2018 10:44:19 PM
Step Count 16 Switch Count 1
143! /* sleep for 2 seconds */
144
145 title5 "Print of Laboratory Analysis Dataset";
146 title6 "** %sysfunc(datetime(),datetime21.) **";
147 proc print data = ADLB; /* t is same as when loading hash for the first time */
148 run;
NOTE: There were 13 observations read from the data set WORK.ADLB.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.01 seconds
user cpu time 0.01 seconds
system cpu time 0.00 seconds
memory 359.68k
OS Memory 36340.00k
Timestamp 10/08/2018 10:44:19 PM
Step Count 17 Switch Count 0
149
150 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

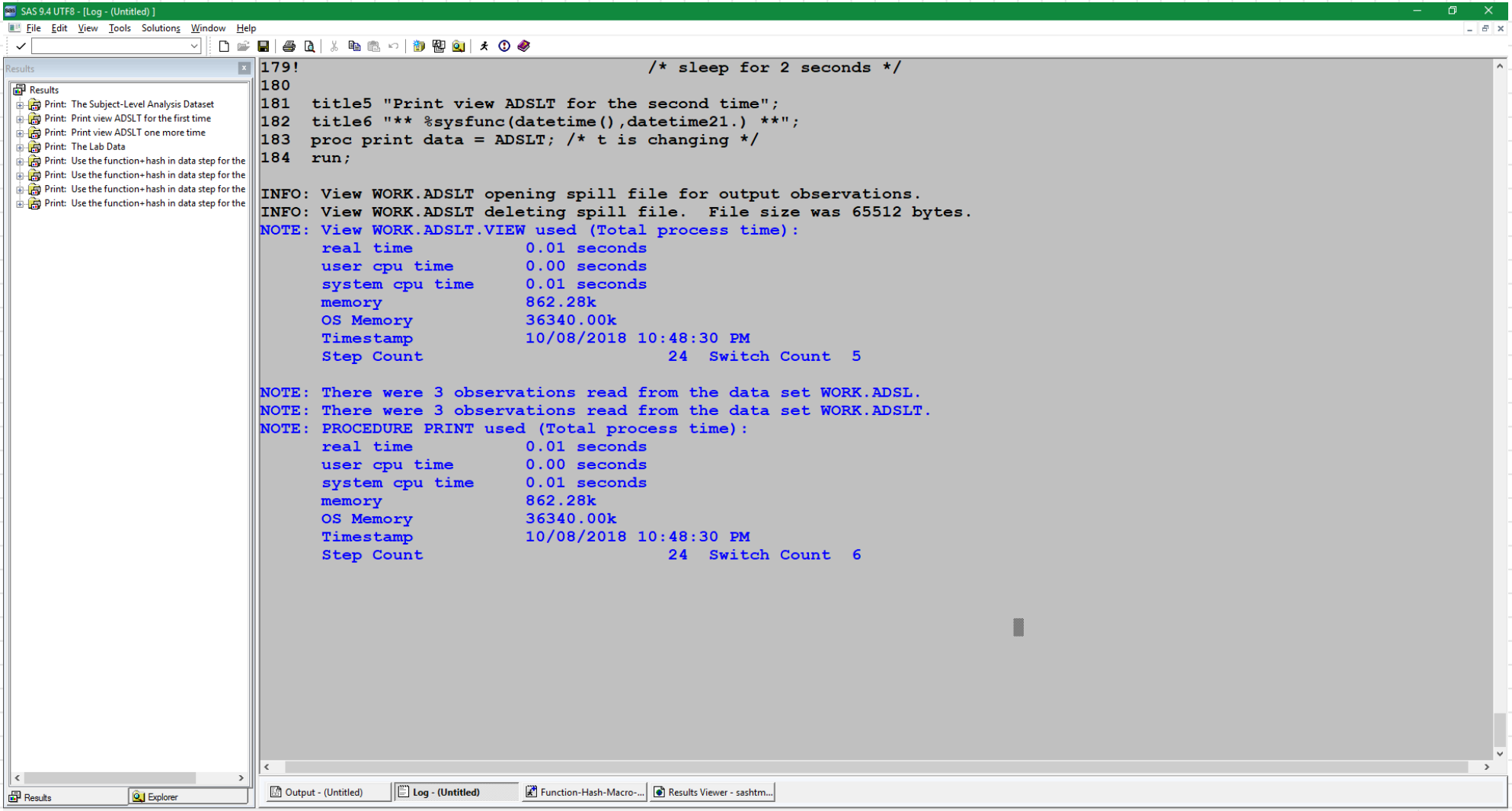

SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 306.09k
OS Memory 36340.00k
Timestamp 10/08/2018 10:44:21 PM
Step Count 18 Switch Count 1
150! /* sleep for 2 seconds */
151
152 title5 "Print view ADSLT for the second time";
153 title6 "** %sysfunc(datetime(),datetime21.) **";
154 proc print data = ADSLT; /* t is changing */
155 run;
INFO: View WORK.ADSLT opening spill file for output observations.
INFO: View WORK.ADSLT deleting spill file. File size was 65512 bytes.
NOTE: View WORK.ADSLT.VIEW used (Total process time):
real time 0.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 866.78k
OS Memory 36340.00k
Timestamp 10/08/2018 10:44:21 PM
Step Count 19 Switch Count 5
NOTE: There were 3 observations read from the data set WORK.ADSL.
NOTE: There were 3 observations read from the data set WORK.ADSLT.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 866.78k
OS Memory 36340.00k
Timestamp 10/08/2018 10:44:21 PM
Step Count 19 Switch Count 6
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK

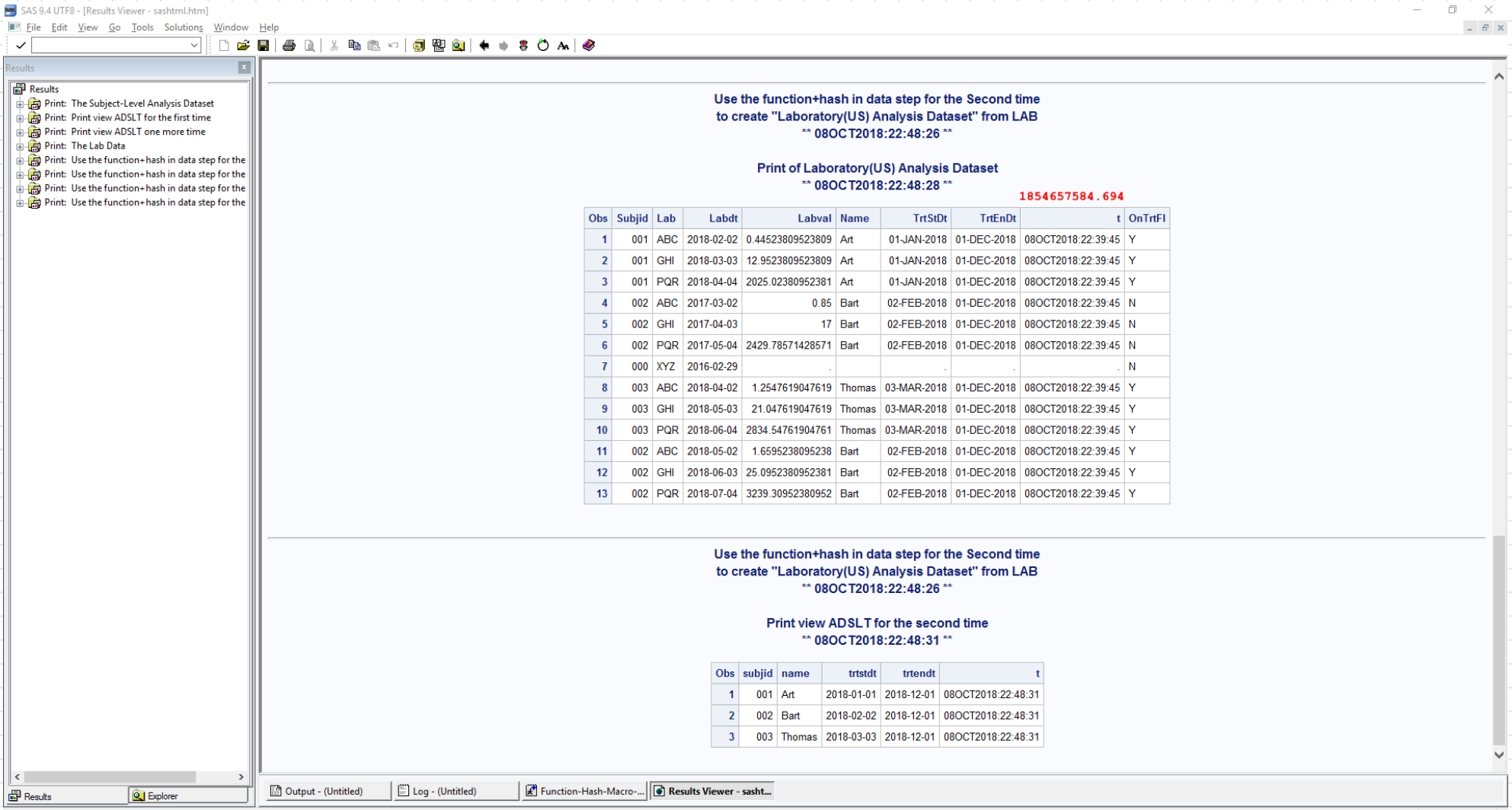

SAS 9.4 UTF8 - [Results Viewer - sashtml.htm]
File Edit View Go Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Use the function+hash in data step for the First time
to create "Laboratory Analysis Dataset" from LAB
** 08OCT2018:22:44:17 **
Print of Laboratory Analysis Dataset
** 08OCT2018:22:44:19 **
1854657584.694
Obs Subjid Lab Labdt Labval Name TrtStDt TrtEnDt t OnTrtFl
1 001 ABC 2018-02-02 1.1 Art 01-JAN-2018 01-DEC-2018 08OCT2018:22:39:45 Y
2 001 GHI 2018-03-03 32 Art 01-JAN-2018 01-DEC-2018 08OCT2018:22:39:45 Y
3 001 PQR 2018-04-04 5003 Art 01-JAN-2018 01-DEC-2018 08OCT2018:22:39:45 Y
4 002 ABC 2017-03-02 2.1 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 N
5 002 GHI 2017-04-03 42 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 N
6 002 PQR 2017-05-04 6003 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 N
7 000 XYZ 2016-02-29 . . . . N
8 003 ABC 2018-04-02 3.1 Thomas 03-MAR-2018 01-DEC-2018 08OCT2018:22:39:45 Y
9 003 GHI 2018-05-03 52 Thomas 03-MAR-2018 01-DEC-2018 08OCT2018:22:39:45 Y
10 003 PQR 2018-06-04 7003 Thomas 03-MAR-2018 01-DEC-2018 08OCT2018:22:39:45 Y
11 002 ABC 2018-05-02 4.1 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 Y
12 002 GHI 2018-06-03 62 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 Y
13 002 PQR 2018-07-04 8003 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 Y
Use the function+hash in data step for the First time
to create "Laboratory Analysis Dataset" from LAB
** 08OCT2018:22:44:17 **
Print view ADSLT for the second time
** 08OCT2018:22:44:21 **
Obs subjid name trtstdt trtendt t
1 001 Art 2018-01-01 2018-12-01 08OCT2018:22:44:21
2 002 Bart 2018-02-02 2018-12-01 08OCT2018:22:44:21
3 003 Thomas 2018-03-03 2018-12-01 08OCT2018:22:44:21
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sasht...
C:\SAS_WORK


SAS 9.4 UTF8 - [Function-Hash-Macro-sandwich.sas *]
File Edit View Tools Run Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
294 title1 "Use the function+hash in data step for the Second time";
295 title2 'to create "Laboratory(US) Analysis Dataset" from LAB';
296 title3 "** %sysfunc(datetime(),datetime21.) **";
297 data ADZL; /* use our function with data for the second time */
298 set LAB;
299 length Name $ 10 TrtStDt TrtEnDt t 8 OnTrtFl $ 1;
300 format trtstdt trtendt date11. t datetime21.;
301 call missing(name, trtstdt, trtendt, t);
302
303 call GetADSL(subjid, name, trtstdt, trtendt, t);
304
305 ontrtfl = ifc(trtstdt <= labdt <= trtendt, "Y", "N");
306
307 labval = (labval * 17) / 42; /* conversion from SI to US ;-) */
308 run;
309
310 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
311
312 title5 "Print of Laboratory(US) Analysis Dataset";
313 title6 "** %sysfunc(datetime(),datetime21.) **";
314 proc print data = ADZL; /* t is same as when loading hash for the first time */
315 run;
316
317 data _null_; rc=sleep(2,1); run; /* sleep for 2 seconds */
318
319 title5 "Print view ADSLT for the second time";
320 title6 "** %sysfunc(datetime(),datetime21.) **";
321 proc print data = ADSLT; /* t is changing */
322 run;
323
324 /*
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK
Ln 300, Col 48


SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
156 title1 "Use the function+hash in data step for the Second time";
157 title2 'to create "Laboratory(US) Analysis Dataset" from LAB';
158 title3 "** %sysfunc(datetime(),datetime21.) **";
159 data ADZL; /* use our function with data for the second time */
160 set LAB;
161 length Name $ 10 TrtStDt TrtEnDt t 8 OnTrtFl $ 1;
162 format trtstdt trtendt date11. t datetime21.;
163 call missing(name, trtstdt, trtendt, t);
164
165 call GetADSL(subjid, name, trtstdt, trtendt, t);
166
167 ontrtfl = ifc(trtstdt <= labdt <= trtendt, "Y", "N");
168
169 labval = (labval * 17) / 42; /* conversion from SI to US ;-) */
170 run;
NOTE: Missing values were generated as a result of performing an operation on missing values.
Each place is given by: (Number of times) at (Line):(Column).
1 at 169:19
NOTE: There were 13 observations read from the data set WORK.LAB.
NOTE: The data set WORK.ADZL has 13 observations and 9 variables.
NOTE: DATA statement used (Total process time):
real time 0.03 seconds
user cpu time 0.01 seconds
system cpu time 0.01 seconds
memory 1420.46k
OS Memory 36340.00k
Timestamp 10/08/2018 10:48:26 PM
Step Count 20 Switch Count 0
171
172 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
173
174 title5 "Print of Laboratory(US) Analysis Dataset";
175 title6 "** %sysfunc(datetime(),datetime21.) **";
176 proc print data = ADZL; /* t is same as when loading hash for the first time */
177 run;
NOTE: There were 13 observations read from the data set WORK.ADZL.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.01 seconds
user cpu time 0.01 seconds
system cpu time 0.00 seconds
memory 359.68k
OS Memory 36340.00k
Timestamp 10/08/2018 10:48:28 PM
Step Count 22 Switch Count 0
178
179 data _null_; rc=sleep(2,1); run;
NOTE: DATA statement used (Total process time):
real time 2.01 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 306.09k
OS Memory 36340.00k
Timestamp 10/08/2018 10:48:30 PM
Step Count 23 Switch Count 1
179! /* sleep for 2 seconds */
180
181 title5 "Print view ADSLT for the second time";
182 title6 "** %sysfunc(datetime(),datetime21.) **";
183 proc print data = ADSLT; /* t is changing */
184 run;
INFO: View WORK.ADSLT opening spill file for output observations.
INFO: View WORK.ADSLT deleting spill file. File size was 65512 bytes
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
179! /* sleep for 2 seconds */
180
181 title5 "Print view ADSLT for the second time";
182 title6 "** %sysfunc(datetime(),datetime21.) **";
183 proc print data = ADSLT; /* t is changing */
184 run;
INFO: View WORK.ADSLT opening spill file for output observations.
INFO: View WORK.ADSLT deleting spill file. File size was 65512 bytes.
NOTE: View WORK.ADSLT.VIEW used (Total process time):
real time 0.01 seconds
user cpu time 0.00 seconds
system cpu time 0.01 seconds
memory 862.28k
OS Memory 36340.00k
Timestamp 10/08/2018 10:48:30 PM
Step Count 24 Switch Count 5
NOTE: There were 3 observations read from the data set WORK.ADSL.
NOTE: There were 3 observations read from the data set WORK.ADSLT.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.01 seconds
user cpu time 0.00 seconds
system cpu time 0.01 seconds
memory 862.28k
OS Memory 36340.00k
Timestamp 10/08/2018 10:48:30 PM
Step Count 24 Switch Count 6
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sashtm...
C:\SAS_WORK


SAS 9.4 UTF8 - [Results Viewer - sashtml.htm]
File Edit View Go Tools Solutions Window Help
Results
Results
Print: The Subject-Level Analysis Dataset
Print: Print view ADSLT for the first time
Print: Print view ADSLT one more time
Print: The Lab Data
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Print: Use the function+hash in data step for the
Use the function+hash in data step for the Second time
to create "Laboratory(US) Analysis Dataset" from LAB
** 08OCT2018:22:48:26 **
Print of Laboratory(US) Analysis Dataset
** 08OCT2018:22:48:28 **
1854657584.694
Obs Subjid Lab Labdt Labval Name TrtStDt TrtEnDt t OnTrtFl
1 001 ABC 2018-02-02 0.44523809523809 Art 01-JAN-2018 01-DEC-2018 08OCT2018:22:39:45 Y
2 001 GHI 2018-03-03 12.9523809523809 Art 01-JAN-2018 01-DEC-2018 08OCT2018:22:39:45 Y
3 001 PQR 2018-04-04 2025.02380952381 Art 01-JAN-2018 01-DEC-2018 08OCT2018:22:39:45 Y
4 002 ABC 2017-03-02 0.85 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 N
5 002 GHI 2017-04-03 17 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 N
6 002 PQR 2017-05-04 2429.78571428571 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 N
7 000 XYZ 2016-02-29 . . . . N
8 003 ABC 2018-04-02 1.2547619047619 Thomas 03-MAR-2018 01-DEC-2018 08OCT2018:22:39:45 Y
9 003 GHI 2018-05-03 21.047619047619 Thomas 03-MAR-2018 01-DEC-2018 08OCT2018:22:39:45 Y
10 003 PQR 2018-06-04 2834.54761904761 Thomas 03-MAR-2018 01-DEC-2018 08OCT2018:22:39:45 Y
11 002 ABC 2018-05-02 1.6595238095238 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 Y
12 002 GHI 2018-06-03 25.0952380952381 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 Y
13 002 PQR 2018-07-04 3239.30952380952 Bart 02-FEB-2018 01-DEC-2018 08OCT2018:22:39:45 Y
Use the function+hash in data step for the Second time
to create "Laboratory(US) Analysis Dataset" from LAB
** 08OCT2018:22:48:26 **
Print view ADSLT for the second time
** 08OCT2018:22:48:31 **
Obs subjid name trtstdt trtendt t
1 001 Art 2018-01-01 2018-12-01 08OCT2018:22:48:31
2 002 Bart 2018-02-02 2018-12-01 08OCT2018:22:48:31
3 003 Thomas 2018-03-03 2018-12-01 08OCT2018:22:48:31
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Function-Hash-Macro-...
Results Viewer - sasht...
C:\SAS_WORK

thank you

references

- Art Carpenter, "Using Hash Tables to Manage Your Macro Language Control Files", SAS GF 2018 Proceedings, https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2399-2018.pdf
- Art Carpenter, "Using Hash Tables to Manage Your Macro Language Control Files" (final), WUSS 2018 Proceedings, https://www.lexjansen.com/wuss/2018/41_Final_Paper_PDF.pdf
- Art Carpenter, "Carpenter's Guide to Innovative SAS Techniques", SAS Press
- Art Carpenter, "Carpenter's Complete Guide to the SAS Macro Language", SAS Press
- Ian Whitlock, "The RESOLVE Function - What Is It Good For?", NESUG 1998 Proceedings, https://www.lexjansen.com/nesug/nesug98/code/p088.pdf
- Paul M. Dorfman, Koen Vyverman, "Data Step Hash Objects as Programming Tools", SUGI 30 Proceedings, www2.sas.com/proceedings/sugi30/236-30.pdf
- Chris Schacherer, "Introduction to SAS Hash Objects", SAS GF 2015 Proceedings, https://support.sas.com/resources/papers/proceedings15/3024-2015.pdf
- Paul M. Dorfman, Don Henderson, "Data Aggregation Using the SAS Hash Object", SAS GF 2015 Proceedings, https://support.sas.com/resources/papers/proceedings15/2000-2015.pdf
- Paul M. Dorfman, "Fundamentals of the The SAS Hash Object", SESUG 2016 Proceedings, https://analytics.ncsu.edu/sesug/2016/HOW-195_Final_PDF.pdf
- Dylan Ellis, "RUN_MACRO Run! With PROC FCMP and the RUN_MACRO Function from SAS 9.2, Your SAS Programs Are All Grown Up", SAS GF 2013 Proceedings, https://support.sas.com/resources/papers/proceedings13/033-2013.pdf
- Andrew Henrick, Donald Erdman, Stacey Christian, "Hashing in PROC FCMP to Enhance Your Productivity", SAS GF 2013 Proceedings, http://support.sas.com/resources/papers/proceedings13/129-2013.pdf
- Mike Rhoads, "Use the Full Power of SAS in Your Function-Style Macros", SAS GF 2012 Proceedings, https://support.sas.com/resources/papers/proceedings12/004-2012.pdf
- Rick Langston, "Submitting SAS Code On The Side", SAS GF 2013 Proceedings, https://support.sas.com/resources/papers/proceedings13/032-2013.pdf