

The Data Review Two-step: Macros for Comparison and Exception Reports

Ronald Fehd, Centers for Disease Control and Prevention, Atlanta GA

Part A: %COMPARWS: Compare with summary: a macro using proc COMPARE to write a file of differences to edit and use for updates

ABSTRACT

Data review can be viewed as a two step process:

1. Compare: review differences between two data sets.
2. Exception Report: review data consistency with range and logic checks.

A data set may updated with a file of the form:

```
* if ID = 1 then do;  
* <variable name> = <value>;  
* end;
```

This paper reviews the proc COMPARE output data set and the manipulation of that data set which is necessary to write a report to a file which contains update statements in an easily editable form. Summary information is written at the end of the update file.

INTRODUCTION

Data review is a necessary step in data management. Its purpose is to provide accurate and consistent data for data analysis. SAS® software provides proc COMPARE as a useful tool to identify differences between two data sets. These data sets may be the result of two independent data entry steps or the results of scanning using different degrees of optical character recognition or one of each of these two processes.

With large survey data sets reviewing, indicating, and making corrections using a printed report from proc COMPARE can be tedious in two areas:

1. Low information per page: many pages in report
2. Transferring corrections from paper to an update file necessitates retyping variable names and values; this, in turn, requires an edit and correct cycle.

This paper reports on a utility macro which fulfills these criteria:

1. Concise report per observation, easy to read and edit.
2. Difference report is written to file for later editing and subsequent use as an %included update file.
3. Summary report provided for data entry process review.

For a discussion of exception reports see Fehd 1997, XRPT.

Assumptions about the data sets to be compared

Pairs of data sets to be used with this macro are expected to have the same structure, identifiers (ID), and number of observations. The test in step 1 checks the ID list and exits if it is blank. The %NOBS macro (Fehd 1997) is based on the %OBSNVAR macro (SAS Macro Language Reference, First Edition). Step 2 uses the %NOBS macro to provide the number of observations of the base and comparison data sets. Step 3 compares the number of observations in both data sets and exits if they are unequal. It is assumed that both data sets are sorted and unique on the IDs, else proc COMPARE will fail.

Specifications of correction file to be used for updates

In order for a text file to be usable as an %included update file, it must be written in the following format:

```
* if ID = 1 then do;          *1 ID;  
* variable = <value>;        *2 variable assignment;  
* end;                       *3 ID closure;
```

1. The ID line opens an if-do statement. This example shows a single numeric id. If the ID is character then the ID value must be enclosed in quotes. Similarly, if other secondary IDs are character, they must likewise be quoted. The first specification is:

1. quote ID values, according to their type.

This is accomplished by using the proc CONTENTS of the base data set and saving the type in a macro variable. Step 4 uses %ARRAY (Fehd 1996) to create macro variables of each of the IDs. In step 5 the type of each ID is saved as a macro variable.

2. This procedure was written to eliminate the errors associated with retyping variable names and values. As with IDs, variables listed must be quoted, or not, according to their type. This information is available in the proc COMPARE output data set and is used in the report writer in step 10. The first part of the variable assignment specification is:

2.1 quote difference values, according to their type.

This report lists two different values of the same variable, one from the base data set, the other from the comparison data set. Visual comparison of short sequences of letters

or digits to locate differences is relatively easy. For sequences longer than five characters it may be necessary to offer a visual indicator of the position of the difference. The second part of the variable assignment specification is:

2.2 for length greater than 5, indicate where difference is

3. Every open control statement must have closure. The last specification is:

3. provide end statement.

These specifications are relatively simple to implement in a data _NULL_ report writer. We now review proc COMPARE and its options with a view to producing a data set which is usable by the report writer.

Proc COMPARE: report and output data set

The prototype for this macro is the proc COMPARE Value Comparison Results report (SAS® Procedures Guide, Version 6, Third Edition). The transpose option is used to report by observation or ID. This report is paged by ID, which produces a minimum of one page per ID, no matter the number of variables reported different.

In step 6 proc COMPARE creates an output data set. The option outall writes four observations per ID with any difference. These are identified by the variable _Type_.

<u>_TYPE_</u>	<u>explanation</u>
BASE	value in base data set
COMPARE	value in compare data set
DIF	difference marker for character
PERCENT	difference of numeric variable

In step 7 the number of observation of the output data set is checked. If there are no differences, a message is written to the output file and the processing is ended.

Step 8: The difference data set is transposed and then made unique on the differences. This process produces the data set for the report writer.

DATA _NULL_ report writer

Step 9: Values used in summary are created.

Step 10: The report writer now has a data set with IDs which can be used with first- and last-ID processing. This partially fulfills specifications 1 and completely fulfills specification 3. Macro Q adds appropriate quotes to the identifiers for the fulfilment of the rest of specification 1.

Variables are identified as either character or numeric by comparing the values of Dif and Percent. Appropriate quotes are provided when writing the detail lines; this fulfills specification 2.1. Specification 2.2 is met by an

optional comment line which prints the Dif variable containing an 'X' at the position of difference.

Summary Information

This report provides several types of summary information. At end of the print file is the following data: the number of IDs, variables and cells in the base and difference data sets, and percentages of each. This data is intended as feedback to the data entry process. The percentage of cells reported different is the data entry error rate.

Example of detail with summary data (from test data). Data entry error rate is at lower right: Pcnt : Cells

```
* if NID1 = 3 then do;
*   D3      = '!' ;
*   D3      = '!' ;
* end;*NDiff =1 ;
***smry :   IDs      Vars      Cells;
*** Base:    4        8        32;
*** Diff:    3        4        4;
*** Pcnt:   75%      50%      12.5%;
```

A list of the differences data set, by variable name, is provided so that errors within a variable may be reviewed. When the report is written to a print file, the following data sets may be enabled to be printed: Difference-IDs, and Difference-Variables.

Usage:

In a production setting this macro is first used to write and print both a correction file and a paper report. The paper report is reviewed by colleagues responsible for data collection. The marked up copy is returned and the desired corrections are enabled in the file by removing the asterisks in front of the appropriate statements. These updates are applied to the data and a third data set created. The macro is then used to write only a paper report, which is used to verify that all indicated corrections were made.

Program COMPAR12: report differences
filename COMPRPT <COMPARWS-destination>;
%COMPARWS(LIBRARY.<DATA1>
,LIBRARY.<DATA2>,IDNMBR
,PRNTFILE=COMPRPT);
note: writes report to file, prints report for review.

Process: review report, note corrections to be made.

Process: edit corrections file, enable corrections.

Program DATA3: apply corrections
DATA <DATA3>;
set <DATA2>;
%include COMPRPT;

Program COMPAR23: confirm corrections
%COMPARWS(LIBRARY.<DATA2>
,LIBRARY.<DATA3>,IDNMBR);

note: print report to review and confirm corrections.

CONCLUSION

Proc COMPARE provides a voluminous report of differences between two data sets. This routine use proc COMPARE's output data set to provide a concise difference report which may be saved as a text file or printed for review. When corrections are necessary the text file is edited instead of retyping variable names and values. Using the text file to update the data set eliminates the edit and correct cycle associated with manually retyping variables and values.

REFERENCES

Fehd, Ronald (1996), "%ARRAY, construction and usage of arrays of macro variables," *Proceedings of the Fourth Annual Conference of the Southeast SAS Users Group*, 4, 156-160.

Fehd, Ronald (1997), "%NOBS: use %sysfunc to return number/obs of data set," *SAS-L archives, contribution 38*.

Fehd, Ronald (1997), "XRPT: macros %ID and %SHOW for writing Exception Reports: perform range and logic checks on a data set; write file of exceptions to edit and use for updates" *Proceedings of the Eighth Annual Conference of the Midwest SAS Users Group*, 8.

SAS Institute Inc. (1997), *SAS® Macro Language*

```

/* *****
* MACRO: COMPARE NOTE: uses macros ARRAY, NOBS *
*
* USAGE: 1. %COMPARE(DATA1,DATA2,IDX);
* 2. %COMPARE(DATA1,DATA2,IDX,PRINTFILE=<fileref>);
* 3. %COMPARE(DATA1,DATA2,IDX,TESTING=1);
* 4. %COMPARE(DATA1,DATA2,IDX,DROPLIST=Var1 Var2 .. VarN);
* recommend drop character variables where length gt 12
* 5. %COMPARE(DATA1,DATA2,IDX,READBASE=XXXX,READCOMP=YYYY);
* 6. TITLE 'title for this project';
* %COMPARE(DATA1,DATA2,IDX,TITLE=2);
* 7. %COMPARE(DATA1,DATA2,IDX,IDX2,IDX2=2);
*
* DESCRIPTION: use proc COMPARE output data set
* to write corex file
* and summary report of differences
*
* PROCESS: 1. if IDXLIST blank print error-msg ..... exit *
* 2. %NOBS of BASE & COMPARE for error-check
* 3. if error-checks print error-msg ..... exit *
* 4. %ARRAY of ID(s)
* 5. vars in IDXLIST (character? numeric?)
* 6. proc COMPARE out = COMPARE
* 7. %NOBS, if no differences print exit-message ... exit
* 8. proc TRANSPOSE, rotate and unique data set
* 9. make values used in summary
* 10. write corex to PRINTFILE
* 11. if wanted print summary report
*
* NOTES: width of variables, in COMPARE output data set
* depends on largest character variable in input data set
* DROPLIST is provided to delete wide variables from comparison
* *!> >>> This routine will fail for large numbers of mismatches *!>
* *!> where data set has one wide character variable *!>
* *!> which increases the size of the COMPARE output data set *!>
*
* KEYWORDS: %ARRAY %NOBS %COMPARE %TRANSPOSE
* corrections updates "output data set from procedure"
*
* Author: Ronald Fehd, B.S. C.80 e-mail: RJF2@cdc.gov
* Centers for Disease Control MS-G25
* 4770 Buford Hwy NE
* Atlanta GA 30341-3724 voice: 770/488-4316
* *****
*/
%MACRO COMPARE( /*
BASE /* BASE for proc COMPARE
COMPARE /* COMPARE for proc COMPARE
** either or both may be 2-level names

```

Reference, First Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS® Procedures Guide, Version 6, Third Edition*, Cary, NC: SAS Institute Inc.

SAS is a registered trademark of SAS Institute, Inc. In the USA and other countries, ® indicates USA registration.

Author: Ronald Fehd e-mail: RJF2@cdc.gov
Centers for Disease Control MS-G25
4770 Buford Hwy NE
Atlanta GA 30341-3724 voice: 770/488-4316
SAS-L archives: send e-mail
to: SAScontrib@SASserv.uga.edu
for %ARRAY subject: cntb0031: download
for %COMPARWS subject: cntb0034: download
for %NOBS subject: cntb0038: download
for XRPT subject: cntb0039: download

ACKNOWLEDGMENTS

These routines were developed over a period of ten years while I crunched the numbers of survey data collected by the Model Performance Evaluation Program (MPEP) of the Division of Laboratory Systems, Public Health Practice Program Office of the Centers for Disease Control, and Prevention, Atlanta, Georgia. John Hancock, chief, Information Services Activity, encouraged me to write up these routines. I wish to thank Sharon Blumer, David Cross, Thomas Hearn, and William Schalla of the MPEP group for their help in developing this routine.

```

,IDXLIST /* list of primary-key(s) == by-vars */
,IDX_IDX=3 /* number of identifiers, changed by NOBS */
,PRINTFILE=PRINT /* output destination, default = PRINT
/* may be 'external-file' or fileref
,DROPLIST /* list of vars to be dropped from comparison
/* recommend drop very wide character variables: len gt 12
,READBASE /* password for BASE
,READCOMP /* password for COMPARE
,TESTING =0 /* testable test msg and prints?
,TITLE /* TITLE line 6
); /* *****
TITLETITLE. "comparison of BASE. and SCOMPARE.";

%*1. if IDXLIST blank, print err-msg, exit;

%IF IDXLIST = %THEN %DO; %put !!!!!!!;
%put !! no ID-List !!;
%put !!!!!!!; %GOTO ENDOMACR; %END;

%*2. %NOBS of BASE & COMPARE for error-check;

%local BASENOBS; %NOBS(BASENOBS,data = BASE );
%local COMPNOS; %NOBS(COMPNOS,data = SCOMPARE);

%*3. if err-checks print err-msg, exit;

%IF BASENOBS ne COMPNOS /*.....*/
or not BASENOBS
or not COMPNOS %THEN %DO; DATA _NULL_;
file SPRINTFILE;
put
%IF BASENOBS ne COMPNOS %THEN "****number of obs differ, "
"BASE. = BASENOBS. "
"SCOMPARE. = COMPNOS.;"
%IF not BASENOBS %THEN "****no obs in BASE.;"
%IF not COMPNOS %THEN "****no obs in SCOMPARE.;"
/*put closure*/ ;
%GOTO ENDOMACR; /*.....*/ %END;

%*4. %ARRAY of ID(s): for use in macro loops;

%local I; %DO I = 1 %TO IDX_IDX; %local IDXEXI.; %END;
%LET IDXLIST = %upcase(%IDXLIST.); /* proc CONTENTS Name is upcase */
%ARRAY(IDS,IDXLIST);

```

```

%5. vars in XLIST (character? numeric?);

proc CONTENTS data = BASE;
  %IF "SHEADBASE" ne "" %THEN (read=SHEADBASE.);
  out = BASECOUNT (keep = Name Type) noprint;

DATA _NULL_;
do until(EndOfFile);
  set BASECOUNT end = EndOfFile; /*.....*/
  *Type == (1=numeric, 2=character) from proc CONTENTS;
  if Type = 1 then C_N = 'N';
  else C_N = 'C';
  %DO I = 1 %TO SDIN_IDX.;
    if Name = "SASID61." then call symput("TYPEX.",C_N); %END;
  /*.....*/ do until(EndOfFile); /* end;
stop;

%6. proc COMPARE
Base = BASE; /*IF "SDROPLIST" ne "" %THEN drop=SDROPLIST.;
%IF "SHEADBASE" ne "" %THEN read=SHEADBASE.;
compare = SCOMPARE; /*IF "SDROPLIST" ne "" %THEN drop=SDROPLIST.;
%IF "SHEADCOMP" ne "" %THEN read=SHEADCOMP.;

out = COMPARE;
outall /* == outbase outcomp outdiff outpercent */
outnoequal /* write only obs with one or more differences */
noprint;
by XLIST.;

%macro TESTPRINT(DATA,MSG); /*.....*/
%IF &TESTING %THEN %DO;
  proc PRINT data=BASE;
    TITLE%val(&TITLEN.-1) "data=BASE: MSGS";
    proc CONTENTS data = BASE;
  /*.....*/ run; TITLE%val(&TITLEN.-1); /*.....*/ %END; %END;
  %TESTPRINT(COMPARE,output from proc COMPARE);

%7. %NOBS, if no differences print exit-msg, exit;

%local ANYDIFF; %NOBS(ANYDIFF);

%IF not &ANYDIFF %THEN %DO; DATA _NULL_;
  file &PRINTFILE;
  put "====no differences:";
  %&OTS ENDMSG; %END;

%8. TRANSPOSE: rotate & unique data to get one obs with one difference;

proc TRANSPOSE data = COMPARE (drop = _Obs_)
  out = COMPARE (drop = _Label_);
  id _Type_;
  by XLIST.;
  var _all_;
  %TESTPRINT(COMPARE,transposed w/all variables);

DATA COMPARE;
do until(EndOfFile);
  set COMPARE end = EndOfFile;
  where Base ne Compare and _Name_ ne 'TYPE';
  output; /*.....*/ do until(EndOfFile); /* end;
stop; %TESTPRINT(COMPARE,transposed w/only differences);

%9. make values used in summary;
proc FREQ data = COMPARE;
  tables
  %IF SDIN_IDX go 2 %THEN /*do cross-tab of IDs */
  %DO I = 2 %TO SDIN_IDX.; * SASID61. %END;
  / list noprint out = DIFFIDS;
  tables _Name_ / list noprint out = DIFFVARS;

%local DIFFIDS; %NOBS(DIFFIDS,data = DIFFIDS);
%local DIFFVARS; %NOBS(DIFFVARS,data = DIFFVARS);
%local DIFFCLS; %NOBS(DIFFCLS,data = COMPARE);
%local BASEVARS; %NOBS(BASEVARS,data = BASECOUNT);

%10. write corex to PRINTFILE;

DATA %IF "SPRINTFILE" eq "PRINT" %THEN _NULL_;
%ELSE COMPARE; /*DATA closure*/;
options pageno=1;
drop B & C1 C2 C3 D1 Percent MDIFF;
length Q $ 1;
retain B -1 MDIFF 0 B 1 /*counter for proc PRINT sum*/;
file &PRINTFILE;
do until(EndOfFile);
  set COMPARE end = EndOfFile;
  by XLIST.;

  if D1 ne Percent then Q = " "; /* == var is numeric */
  else Q = " "; /* == var is char */

  %MACRO Q(N); /*returns quote for character variables;
    parm is reference */
  %IF &TYPEX. = C %THEN " " /*.....*/ Q. /* %END;

  if first.SASID61_IDX then do; MDIFF = 0; /*.....*/
    put " " if &SD1 = " %Q(1) &SD1 +B %Q(1)
  %IF &SDIN_IDX. go 2 %THEN
    %DO I = 2 %TO SDIN_IDX.; " and SASID61 = " %Q(2) SASID61 +B %Q(2)
    %END;

    then do; MDIFF + 1; /*.....*/ if first. %IF "SPRINTFILE" ne "PRINT" %THEN %DO; output; %END;
    put " " _Name_ MDIFF " " MDIFF +B Base +B Q " ";
    / " " _Name_ MDIFF " " MDIFF +B Compare +B Q " ";
  if length(D1) gt 5
    then put " " diffname MDIFF " " MDIFF +B D1 +B Q " ";
    /*.....*/ if Base ne Compare /* end;

  if last.SASID61_IDX then put " " end; MDIFF = MDIFF " ";
  %IF "SPRINTFILE" ne "PRINT" %THEN %DO; output; %END;
  /*.....*/ do until(EndOfFile); /* end;
  %macro MAKEPNT(VAR,NUM,DENOM); /*create %, trim value to 20 decimals */
  %local D I TRIMLEN; %LET D = 2; %LET TRIMLEN = 0;
  %LET EVAR = %sysval(100 * &NUM / &DENOM);
  %LET I = %index(&EVAR,.) /*if dot in value trim number of decimals;
  %IF &I %THEN %DO; %LET TRIMLEN = %val(%length(&EVAR) - &I);
    %IF &TRIMLEN gt 20 %THEN %LET TRIMLEN = 20;
    %LET EVAR = %substr(&EVAR,1,&I + &TRIMLEN); %END;
  %PUT EVAR = &EVAR; /*.....*/ %END;
  %local BASECLS; %LET BASECLS = %val(&BASENOBS * &BASEVARS);
  %local PCHTNOBS; %MAKEPNT(PCHTNOBS,DIFFIDS, &BASENOBS);
  %local PCHTVARS; %MAKEPNT(PCHTVARS,DIFFVARS, &BASEVARS);
  %local PCHTCLS; %MAKEPNT(PCHTCLS,DIFFCLS, &BASECLS);
  retain C1 12 C2 22 C3 32;
  put "====" C1 "IDs" C2 "Vars" C3 "Cells";
  put "Base:" C1 "BASENOBS" C2 "BASEVARS" C3 "BASECLS";
  put "Diff:" C1 "DIFFIDS" C2 "DIFFVARS" C3 "DIFFCLS";
  put "Pnt:" C1 "PCHTNOBS" C2 "PCHTVARS" C3 "PCHTCLS";
  stop;

  %IF "SPRINTFILE" ne "PRINT" %THEN %DO; /*.....*/
  %11. print summary report; options pageno=1; /*.....*/ cut open *

  proc PRINT data = DIFFIDS;
  TITLE%val(&TITLEN.-1) "summary: IDs listed";
  sum Count;

  proc PRINT data = DIFFVARS;
  TITLE%val(&TITLEN.-1) "summary: Variables listed";
  sum Count;

  proc FREQ data = COMPARE;
  TITLE%val(&TITLEN.-1) "summary: Compare * Base";
  tables Compare * Base / list missing; /*.....*/ cut end */

  proc SORT data = COMPARE;
  by _Name_ Base Compare XLIST.;

  proc PRINT data = COMPARE;
  TITLE%val(&TITLEN.-1) "summary report";
  by _Name_;
  id _Name_;
  sum B ;

  data _NULL_; options pageno=1; /*print detail list*/
  TITLE%val(&TITLEN.-1);
  file PRINT;
  do until(EndOfFile);
    infile &PRINTFILE end = EndOfFile pad lrecl = 72;
    input B1 Line Schar72.;
    put B1 Line Schar72.; /*do until(EndOfFile)/* end;
  stop; /*.....*/ %IF "SPRINTFILE" ne "PRINT" /* %END;
  %ENDMSG; run; TITLE%val(&TITLEN.-1); /*.....*/ %END;
  /* test data section: enable by ending this line with slash (/) -----
  DATA A B; input B1 MD1 1. B1 C1D1 SCHAR1.
    B2 MD2 1. B2 C1D2 SCHAR1.
    B3 B1 3. B3 D1 SCHAR22.
    B3 B2 SCHAR22. B3 B3 SCHAR1.; output A;
  if ID2 = '0' then substr(D2,21,1) = 'A';
  if ID2 = '2' then do; substr(D1,21,1) = 'B';
    N1 = 124; end;
  if ID2 = '3' then D3 = '1';
  output B; cards;
  11123.....
  12123.....
  20123.....
  33323.....
  /*test usage: enable by ending this line with slash (/) -----
  options mprint nocenter;
  *COMPARE(A,B, ) /* test error when no by-list;
  *COMPARE(A,B,MD1); /*test numeric ID;
  *COMPARE(A,B,MD1,PRINTFILE=SASAUTOS(ZCOMPARE)); /*&psys=SYS;

  *filename ZCOMPARE 'c:\sas\zcompare.txt'; /*&psys=SYS;
  *COMPARE(A,B,MD1,PRINTFILE=ZCOMPARE);

  *COMPARE(A,B,MD1,TESTING=1); /*show intermediate data sets;
  *COMPARE(A,B,C1D1); /*test character ID;
  *COMPARE(A,B,MD1 ID2 MD2); /*3 IDs;
  *COMPARE(A,B,MD1 ID2);
  *COMPARE(A,B,MD1 MD2);
  /*example usage of file created by this program as an update file -----
  *data NEW;
  * set OLD;
  * include <file> / source2;
  /*.....*/

```

Part B: XRPT: macros %ID and %SHOW for writing Exception Reports: perform range and logic checks on a data set; write file of exceptions to edit and use for updates

ABSTRACT

Data review can be viewed as a two step process:

1. Compare: review differences between two data sets.
2. Exception Report: review data consistency with range and logic checks.

A data set may be updated with a file of the form:

```
* if ID = 1 then do;  
* <variable name> = <value>;  
* end;
```

This paper reviews common problems in writing exception reports for range and logic checking; the product is a file which contains update commands in an easily editable form. Summary information is written at the end of the update file.

INTRODUCTION

Data review is a necessary step in data management. Its purpose is to provide accurate and consistent data for data analysis. While a comparison report (see Fehd 1997, COMPARWS) may identify individual characters or numbers to be updated, the second half of data review consists of identifying values that are out of the range of expected values for a particular variable.

SAS® software provides procedures which may be used to identify outliers. Proc FREQ may be used for character variables; proc UNIVARIATE may be used for numerics.

This paper reports on a demonstration program and template which fulfills these criteria:

1. Concise exceptions report, per observation, easy to read and edit.
2. Exception report is written to file for later editing and subsequent use as an %included update file.
3. Summary report provided for data processing review.

The primary macros are:

```
%ID: write ID line only once,  
      write message to reviewer  
%SHOW: write variable name and value,  
        using either variable name or array reference
```

Other macros are provided which facilitate quick, clear and easy writing of code to perform range checks of either numeric or character data; data may be checked as individual variables or arrays.

Exception reports are unique to a data set and its variables. The test data set provided contains identifiers, individual variables and arrays in both character and

numeric variables. The specifications may be tested by assigning either character or numeric variables to macro variables used as program parameters. A macro variable TESTING is provided in order to be able to test any new routines.

Specifications of correction file to be used for updates

In order for a text file to be usable as an %included update file, it must be written in the following format:

```
* if ID = 1 then do;          *1 ID;  
* <exception note>;          *2 message;  
* variable = <value>;        *3 variable;  
* end;                        *4 ID closure;
```

1. The ID line opens an if-do statement for an observation with exceptions. This line is to be printed only once. This example shows a single numeric ID. If the ID is character then the ID value must be enclosed in quotes. Similarly, if other secondary identifiers are character, they must likewise be quoted. The first specification is:

- 1.1. *print ID line once per observation.*
- 1.2. *quote ID value(s), according to their type.*

2. The exception message indicates why the variable is being reviewed. The second specification is:

2. *print the exception message as a comment statement.*

3. This program was written to eliminate the errors associated with retyping variable names and values. As with identifiers, variables listed must be quoted, or not, according to their type. The variable assignment specification is:

3. *quote exception values, according to their type.*

4. Every open control statement must have closure. The last specification is:

4. *provide end statement.*

These specifications are implemented in a report writer in step 4.

This XRPTDEMO program contains five sections. The program may be run with the data provided or test data may be inserted into another data set in order to test other macros that call ID and SHOW.

1. Program parameters

TESTING: This macro variable is provided for three purposes: 1. When testing the report is written to file PRINT for quick programmer review. 2. The data set is subset by reducing the number of observations. 3. Invalid values may be inserted into the data by the TESTDATA macro.

ID1, ID2, ID3: Ids are the identifiers or keys of the data set. An exception report is unique to a data set. To add a third ID it is only necessary to enable the macro variable, one line in the ID macro and one line in the summary listings.

DATA_SET: The name of the data set is provided as a macro variable to be used in title and set statements; this ensures that the data evaluated is the correct version.

TITLEn is the number of the last title statement used. It is used to correctly number title lines in the summary.

PRNTFILE is the name of the output file. This file is expected to be edited, have corrections enabled, then used as an %include file for an update. See usage below.

2. Provide variable type to print methods

Macros ID and SHOW are described in object-oriented programming conventions as methods. For these methods to be able to quote character variables the variable type is supplied by creating macro variables from a proc CONTENTS of the data set.

3. Macro definitions

Q returns a single quote for character variables. It is used by both ID and SHOW.

MAKARRAY is used to declare an array of character or numeric variables. It provides the array type to SHOW by declaring a macro variable named with the array name.

ID prints two lines: 1. If the identifier line has not been printed, i.e., if this is the first exception line for an observation, then the line is printed. 2. The exception message is always printed.

SHOW prints a variable and its value, appropriately quoted. The parameter VAR may be either a variable name or an array reference. Where the variable type has been supplied by proc CONTENTS in step 2, the type of an array is supplied by the MAKARRAY macro in step 4.3.

Macros ID and SHOW are used as a pair in this sequence:

```
if <condition> then do;
  %ID(exception message);
  %SHOW(<variable-1> or <array reference>);
  %SHOW(<variable-2>);
end;
```

When a condition refers to two or more variables each

variable is to be shown.

The CHK* macros:

To further reduce the number of lines of code in step 4.4, the programmer-written section of the report writer, three examples using ID and SHOW are provided. These examples illustrate the use of different conditions.

CHKARRAY illustrates reviewing an array.

CHKONE is used for individual character variables with a defined set of values.

CHKHIGH may be used with either character or numeric variables to check if the value exceeds a high point. High points for numeric variables may be discovered by using proc UNIVARIATE. Proc FREQ can be used to identify cutoffs for character variables.

4. Report Writer:

4.1. Setup: The data set is saved for use by the summary listings in step 5. The retained variables are used in the brief summary in step 4.5.

4.2. %TESTDATA: changing values to invalid: Test data has been provided with this demonstration program. To facilitate testing this program as it is written and debugged for other data sets the TESTDATA macro: 1. writes output to file PRINT; 2. reduces the number of observations; 3. provides opportunity to insert dummy values.

4.3. array declarations: If CHKARRAY is to be used arrays must be declared with MAKARRAY.

4.4. list of exceptions: The simple part begins here: for each variable it is only necessary to provide the checking macro, variable- or array-name and check-values. Each of the checking macros generates a minimum of 16 statements. Compare the savings in time spent writing and reviewing one page of code with 50 lines for 50 variables or $50 * 16 = 800$ lines: approximately 16 pages of code.

4.5. ID and data step closure, brief summary: This report provides a brief summary at the end of the print file: the number of observations, variables and cells in the data set, the number of identifiers of the exception data, the number of observations of the exception data set, and percentages of each. This data is intended as feedback to the data collection design process. The percentage of cells reported is the exception report rate.

Example of detail with summary data (from test data).
Exception report rate is at lower right: Pcnt : Cells

```
* if NID1 = 3 then do;
*   XRPT MSG;
*   D3      = ' ' ;
*   end;*XrptInId =1 ;
***smry :  Ids      Vars      Cells;
*** Base:   4        8        32;
*** Diff:   3        4        4;
*** Pcnt:   75%      12.5%;
```

5. Summary listings

When the report is written to a print file, the following data sets may be enabled to be printed: Exception-Identifiers, and Exception-Variables.

A list of the exception data set, by variable name, is provided so that errors within a variable may be reviewed.

Usage:

In a production environment, an exception report program is used to write and print both a correction file and a paper report. The paper report is reviewed by colleagues responsible for data collection. The marked up copy is returned and the desired corrections are enabled in the file by removing the asterisks in front of the appropriate statements. These updates are applied to the data and another data set created. A comparison report (Fehd 1997, COMPARWS) is used to verify that all indicated corrections were made.

Program XRPT
filename XRPT<filespec>;
note: writes report to file, print output for review.

Process: review report, note corrections to be made.

Process: edit corrections file, enable corrections.

Program UPDATE4: apply corrections
DATA <UPDATE4>;
set <DATA3>;
%include XRPT;

Program COMPAR34: confirm corrections
%COMPARWS(LIBRARY.<DATA3>
,LIBRARY.<UPDATE4>
,IDNMBR);
note: print output for review and confirm update.

```
/* *****
* PROGRAM: XRPTDMS: Exception Report Demo, with macros ID and SHOW
* macros: 1 : returns single quote for character vars
*          MAKARAY: declare array, save type of array in mac-var
*          ID : prints ID line once, exception message
*          SHOW : prints variable or array-reference
*          CHKARRAY: check each value in array
*          CHKONE : check one variable is in range
*          CHKHIGH : check one variable is less than high
* DESCRIPTION: data _NULL_ report writer reads data set
*              writes correction file with brief summary
*              writes summary listings of exceptions
* PROCESS: 0. make test data from cards
*          1. set program parameters
*          2. provide variable type, in (N,C), to print methods
*          3. macro definitions
*          4. data step: report writer
*          4.1. initialize vars
*          4.2. if testing, change values to invalid
*          ***programmer-written section ***** begin
*          4.3. array declaration(s), if any
*          4.4. list of exceptions
*          ***programmer-written section ***** end
*          4.5 data step closure, brief summary
*          5. summary listing(s)
* NOTES: ID and SHOW are methods for printing variables
*        variable type is made available to these methods
*        thru mac-vars created from proc CONTENTS data
***** */
```

CONCLUSION

Many different procedures may be used to identify data that is not in the expected set of values for a particular variable. This procedure aids the data review process by writing exceptions to a file where corrections can be made by removing asterisks. This eliminates retyping of variables with their values and the associated edit and correct cycle.

REFERENCES

Fehd, Ronald (1997), "COMPARWS" *Proceedings of the Eighth Annual Conference of the MidWest SAS Users Group*.

SAS is a registered trademark of SAS Institute, Inc. In the USA and other countries. © indicates USA registration.

Author: Ronald Fehd e-mail: RJF2@cdc.gov
Centers for Disease Control MS-G25
4770 Buford Hwy NE voice: 770/488-4316
Atlanta GA 30341-3724
SAS-L archives: send e-mail
to: SAScontrib@SASserv.uga.edu
for %COMPARWS subject: cntb0034: download
for XRPT subject: cntb0039: download

ACKNOWLEDGMENTS

These routines were developed over a period of five years while I crunched the numbers of sample and questionnaire survey data collected by the Model Performance Evaluation Program (MPEP) of the Division of Laboratory Systems, Public Health Practice Program Office of the Centers for Disease Control, and Prevention, Atlanta, Georgia. John Hancock, chief, Information Services Activity, encouraged me to write up these routines. I wish to thank Sharon Blumer, David Cross, Thomas Hearn, and William Schalla of the MPEP group for their data collection.

```
***** or mac-vars created by MAKARAY *****
* KEYWORDS: "data _NULL_ report writer" "exception report"
* "object methods"
* Author: Ronald Fehd, B.S. O.Sc e-mail: RJF2@cdc.gov
* Centers for Disease Control MS-G25
* 4770 Buford Hwy NE
* Atlanta GA 30341-3724 voice: 770/488-4316
***** */
/*%0. TEST DATA disable after first run *****
data XRPTDMS; infile cards;
input 01 CID Schar2. 01 MID 2.
      03 CID2 Schar1. 03 MID2 1.
      04 C1 Schar1. 05 M1 1.
      06 C21 Schar1. 07 C22 Schar1. 08 C23 Schar1.
      09 M21 1. 07 M22 1. 08 M23 1.
      030 C3 Schar2. 030 M3 2.
      030 C4 Schar3. 030 M4 3. ;cards;

011A1010101000 010
021A1010110101 120
031A1010101010 341
032A1010101010 341
/*4567890123456789012345678901234567890
1 2 3 4 5*****
%*1. set program parameters;
%LET TESTING = 1; /*set output file to PRINT, enable dummy TESTDATA*/
%LET TESTING = 0;
%LET ID1 = CID; /*id is character*/
%LET ID1 = MID; /*id is numeric */
```

```

%LET ID2 = /*secondary id is blank*/
%LET ID2 = &ID2;
%LET ID2 = &ID2;
%LET ID3 = /*tertiary id is not enabled*/

%LET DATA_SET = XNPTDMS; /*test data provided*/
%LET DATA_SET = LIBRARY.XNPTDATA;

TITLE1 'XNPTDMS: Exception Report: Data out of range or inconsistent';
TITLE2 'report from &DATA_SET.';
%LET TITLEN = 2; /*used for numbering titles in summary listings*/

%LET PRINTFILE = 'C:\BAS\XNPT.SAS';
%LET PRINTFILE = PRINT;

options linesize = 85 pagesize = 54; SAS ms 10 portrait duplex; long;
options /NO/ mprint nocenter pagesno=1;

/*SETUP begin !!!!!!!!!!!!!!! RUN THIS SECTION ONCE !!!!!!!!!!!!!!!*/
%*2. provide variable type, in (N,C), to print methods;
proc CONTENTS data = &DATA_SET
    out = XNPTCONT (keep = Name Type Nobs) noprint;
data _NULL_;
length NC $ 1;
do until (EndOfFile);
    set XNPTCONT end = EndOfFile nobs=Nvars;
    if Type = 1 then NC = 'N';
    else if Type = 2 then NC = 'C';
    call symput (Name,NC);
    call symput ('DATANOB',trim(left(put(Nobs,8)))));
    call symput ('DATAVAR',trim(left(put(Nvars,8)))));
stop; run; /****** SETUP end*/

%*3. macro definitions;
%macro &(TYPE); /*returns quote for character variables; parm in value */
%IF %TYPE = "C" %THEN %DO; /*******/
%END;

%macro MAKARRAY (ARRAYNAME,C,N,ELEMENTS); /* provide array type *****/
%global NAME; %LET NAME = &ARRAYNAME;
%global SHAME; %LET SHAME = &C.N.;
array &ARRAYNAME (*) %IF "C" %THEN "C" %THEN $; ELEMENTS;
%IF %TESTING %THEN %PUT TYPE (NAME) = &SHAME; /****** MAKARRAY */ %END;

%macro ID (NBS); /* print id line, nbs, returns 7 statements *****/
if not ID then do;
    if lyxleft le 4 then put _page_;
    put " " if &ID1 = " " %(&ID1) &ID1 +1 %(&ID1)
        " " and &ID2 = " " %(&ID2) &ID2 +1 %(&ID2);
    %*F &ID3 = " " %(&ID3) &ID3 +1 %(&ID3);
    " then do;";
    Xrptids +1; ID = 1; /*if not ID*/ end;
put " " SHMN; /*******/ %END;

%macro SHOW (VAR); /* print var or array-ref, returns 7 statements *****/
VAR: variable name : TYPE is supplied by mac-vars from proc CONTENTS
    VAR-name is direct reference
    array reference: array-name{I}
    TYPE is supplied by mac-vars from MAKARRAY
    VAR-name comes from vname ******/

%local TYPE C_BRACKT;
%LET C_BRACKT = %index (VAR,{}); /*resolve array-name for type */
%IF %C_BRACKT %THEN %DO; %LET TYPE = %SAS%substr (VAR,1,%C_BRACKT-1);
    call vname (VAR,Name); %END;
%ELSE %DO; %LET TYPE = %SASVAR.;
    Name = "VAR."; %END;

%IF %TESTING %THEN %PUT VAR = <VAR> TYPE = <TYPE>;
put %1 " " %4 Name %13 " " %(&TYPE) VAR. +1 %(&TYPE) ";";
%IF %TYPE = "C" %THEN %DO;
    ValueNum = . ;
    ValueChr = VAR.; %END;
%ELSE %IF %TYPE = "N" %THEN %DO;
    ValueNum = VAR.; %END;
    ValueChr = . ;

Xrptcols + 1; Xrpts + 1; output XNPT; /*******/ %END;

%macro CHKARRAY (NAME) /* returns 2*7+7+2=16 statements *****/
,VALUE=SHMN('D','I',' ',' '));
do I = 1 to dim (NAME.);
    if not ( SHAME {I} in (VALUE.))
        then do;
            %ID (array SHAME not in VALUE)
            %SHOW (SHAME {I}) end;
        /*do I=1:dim*/ end;
%END;

%macro CHKONE (R,RANGE); /* returns 1*7+7+1=16 statements *****/
if verify (R, " . SHAME ") /* NOTE SPACE OR DOT */
    then do;
        %ID (chkone: out of range <SHAME>)
        %SHOW (R); end;
/****** CHKONE */ %END;

%macro CHKIDN (VAR,N); /* returns 1*7+7+1=16 statements *****/
%LET TYPE = %SASVAR.;
if

```

```

%IF %TYPE = "C" %THEN (SAS no " and (SAS go "SHI"));
%ELSE %IF %TYPE = "N" %THEN (SAS no . and (SAS go "SHI"));
then do;
    %ID (review high: VAR. go SHI)
    %SHOW (VAR.) end;
/****** CHKIDN */ %END;

%*4.1. initialize report writer vars;
DATA XNPT (keep = &ID1 &ID2 /*&ID3*/ Name ValueChr ValueNum N);
B1=1; /*pointer control*/
length Name $ 8 ValueChr $ 6 ValueNum $ Xrpts 4 ID 2;
retain Xrptids /*incr by ID, used in brief summary */
    Xrptcols 0 /*incr by SHOW, used in brief summary */
    N 1; /*no change, used in summary listings*/
file &PRINTFILE linesleft = lyxleft;
do until (EndOfFile); /*******/
    set &DATA_SET end = EndOfFile;
    Xrpts = 0; /*incr by SHOW*/
    ID = 0; /*reset by ID */

%*4.2. if testing, change values to invalid;
%macro TESTDATA; /*******/
%IF %TESTING %THEN %DO;
    file PRINT;
    %IF &ID1 = C %THEN %LET ID_VALUE = '02';
    %ELSE %LET ID_VALUE = 2;
    where &ID1 is &ID_VALUE.;
    if &ID1 = &ID_VALUE then do;
        %CHKONE (C1,AS); C1 = 'Z';
        %CHKARRAY (C2); C2 = 'Z';
    /****** TESTDATA */ end; %END; %END; %TESTDATA;

%*programmer-written section ***** begin;
%*4.3. array declaration(s), if any;
%MAKARRAY (C2,C,C2);
%MAKARRAY (N2,N,N2);

%*4.4. list of exceptions;
%CHKONE (C1,AS);
%CHKARRAY (C2);
%CHKARRAY (N2,VALUE=SHMN(0,1));
%CHKIDN (C3,800);
%CHKIDN (N3,800);
%CHKONE (C4,012345678);
%CHKONE (N4,012345);
%*programmer-written section ***** end;

%*4.5 data stop closure, brief summary;
if ID then put " end; " Xrpts Xrpts " &ID1 " &ID1 ";
/******do until (EndOfFile)*/ end;
%LET DATACOLS = %eval (&DATANOB + &DATAVAR);
PcntNobs = 100 * Xrptids / &DATANOB;
PcntCols = 100 * Xrptcols / &DATACOLS;
format PcntNobs PcntCols 7.3;
retain Z1 Z2 Z3 Z4 Z5 Z6 Z7 Z8 Z9 Z10 Z11 Z12 Z13 Z14 Z15 Z16 Z17 Z18 Z19 Z20 Z21 Z22 Z23 Z24 Z25 Z26 Z27 Z28 Z29 Z30 Z31 Z32 Z33 Z34 Z35 Z36 Z37 Z38 Z39 Z40 Z41 Z42 Z43 Z44 Z45 Z46 Z47 Z48 Z49 Z50 Z51 Z52 Z53 Z54 Z55 Z56 Z57 Z58 Z59 Z60 Z61 Z62 Z63 Z64 Z65 Z66 Z67 Z68 Z69 Z70 Z71 Z72 Z73 Z74 Z75 Z76 Z77 Z78 Z79 Z80 Z81 Z82 Z83 Z84 Z85 Z86 Z87 Z88 Z89 Z90 Z91 Z92 Z93 Z94 Z95 Z96 Z97 Z98 Z99 Z100 Z101 Z102 Z103 Z104 Z105 Z106 Z107 Z108 Z109 Z110 Z111 Z112 Z113 Z114 Z115 Z116 Z117 Z118 Z119 Z120 Z121 Z122 Z123 Z124 Z125 Z126 Z127 Z128 Z129 Z130 Z131 Z132 Z133 Z134 Z135 Z136 Z137 Z138 Z139 Z140 Z141 Z142 Z143 Z144 Z145 Z146 Z147 Z148 Z149 Z150 Z151 Z152 Z153 Z154 Z155 Z156 Z157 Z158 Z159 Z160 Z161 Z162 Z163 Z164 Z165 Z166 Z167 Z168 Z169 Z170 Z171 Z172 Z173 Z174 Z175 Z176 Z177 Z178 Z179 Z180 Z181 Z182 Z183 Z184 Z185 Z186 Z187 Z188 Z189 Z190 Z191 Z192 Z193 Z194 Z195 Z196 Z197 Z198 Z199 Z200 Z201 Z202 Z203 Z204 Z205 Z206 Z207 Z208 Z209 Z210 Z211 Z212 Z213 Z214 Z215 Z216 Z217 Z218 Z219 Z220 Z221 Z222 Z223 Z224 Z225 Z226 Z227 Z228 Z229 Z230 Z231 Z232 Z233 Z234 Z235 Z236 Z237 Z238 Z239 Z240 Z241 Z242 Z243 Z244 Z245 Z246 Z247 Z248 Z249 Z250 Z251 Z252 Z253 Z254 Z255 Z256 Z257 Z258 Z259 Z260 Z261 Z262 Z263 Z264 Z265 Z266 Z267 Z268 Z269 Z270 Z271 Z272 Z273 Z274 Z275 Z276 Z277 Z278 Z279 Z280 Z281 Z282 Z283 Z284 Z285 Z286 Z287 Z288 Z289 Z290 Z291 Z292 Z293 Z294 Z295 Z296 Z297 Z298 Z299 Z300 Z301 Z302 Z303 Z304 Z305 Z306 Z307 Z308 Z309 Z310 Z311 Z312 Z313 Z314 Z315 Z316 Z317 Z318 Z319 Z320 Z321 Z322 Z323 Z324 Z325 Z326 Z327 Z328 Z329 Z330 Z331 Z332 Z333 Z334 Z335 Z336 Z337 Z338 Z339 Z340 Z341 Z342 Z343 Z344 Z345 Z346 Z347 Z348 Z349 Z350 Z351 Z352 Z353 Z354 Z355 Z356 Z357 Z358 Z359 Z360 Z361 Z362 Z363 Z364 Z365 Z366 Z367 Z368 Z369 Z370 Z371 Z372 Z373 Z374 Z375 Z376 Z377 Z378 Z379 Z380 Z381 Z382 Z383 Z384 Z385 Z386 Z387 Z388 Z389 Z390 Z391 Z392 Z393 Z394 Z395 Z396 Z397 Z398 Z399 Z400 Z401 Z402 Z403 Z404 Z405 Z406 Z407 Z408 Z409 Z410 Z411 Z412 Z413 Z414 Z415 Z416 Z417 Z418 Z419 Z420 Z421 Z422 Z423 Z424 Z425 Z426 Z427 Z428 Z429 Z430 Z431 Z432 Z433 Z434 Z435 Z436 Z437 Z438 Z439 Z440 Z441 Z442 Z443 Z444 Z445 Z446 Z447 Z448 Z449 Z450 Z451 Z452 Z453 Z454 Z455 Z456 Z457 Z458 Z459 Z460 Z461 Z462 Z463 Z464 Z465 Z466 Z467 Z468 Z469 Z470 Z471 Z472 Z473 Z474 Z475 Z476 Z477 Z478 Z479 Z480 Z481 Z482 Z483 Z484 Z485 Z486 Z487 Z488 Z489 Z490 Z491 Z492 Z493 Z494 Z495 Z496 Z497 Z498 Z499 Z500 Z501 Z502 Z503 Z504 Z505 Z506 Z507 Z508 Z509 Z510 Z511 Z512 Z513 Z514 Z515 Z516 Z517 Z518 Z519 Z520 Z521 Z522 Z523 Z524 Z525 Z526 Z527 Z528 Z529 Z530 Z531 Z532 Z533 Z534 Z535 Z536 Z537 Z538 Z539 Z540 Z541 Z542 Z543 Z544 Z545 Z546 Z547 Z548 Z549 Z550 Z551 Z552 Z553 Z554 Z555 Z556 Z557 Z558 Z559 Z560 Z561 Z562 Z563 Z564 Z565 Z566 Z567 Z568 Z569 Z570 Z571 Z572 Z573 Z574 Z575 Z576 Z577 Z578 Z579 Z580 Z581 Z582 Z583 Z584 Z585 Z586 Z587 Z588 Z589 Z590 Z591 Z592 Z593 Z594 Z595 Z596 Z597 Z598 Z599 Z600 Z601 Z602 Z603 Z604 Z605 Z606 Z607 Z608 Z609 Z610 Z611 Z612 Z613 Z614 Z615 Z616 Z617 Z618 Z619 Z620 Z621 Z622 Z623 Z624 Z625 Z626 Z627 Z628 Z629 Z630 Z631 Z632 Z633 Z634 Z635 Z636 Z637 Z638 Z639 Z640 Z641 Z642 Z643 Z644 Z645 Z646 Z647 Z648 Z649 Z650 Z651 Z652 Z653 Z654 Z655 Z656 Z657 Z658 Z659 Z660 Z661 Z662 Z663 Z664 Z665 Z666 Z667 Z668 Z669 Z670 Z671 Z672 Z673 Z674 Z675 Z676 Z677 Z678 Z679 Z680 Z681 Z682 Z683 Z684 Z685 Z686 Z687 Z688 Z689 Z690 Z691 Z692 Z693 Z694 Z695 Z696 Z697 Z698 Z699 Z700 Z701 Z702 Z703 Z704 Z705 Z706 Z707 Z708 Z709 Z710 Z711 Z712 Z713 Z714 Z715 Z716 Z717 Z718 Z719 Z720 Z721 Z722 Z723 Z724 Z725 Z726 Z727 Z728 Z729 Z730 Z731 Z732 Z733 Z734 Z735 Z736 Z737 Z738 Z739 Z740 Z741 Z742 Z743 Z744 Z745 Z746 Z747 Z748 Z749 Z750 Z751 Z752 Z753 Z754 Z755 Z756 Z757 Z758 Z759 Z760 Z761 Z762 Z763 Z764 Z765 Z766 Z767 Z768 Z769 Z770 Z771 Z772 Z773 Z774 Z775 Z776 Z777 Z778 Z779 Z780 Z781 Z782 Z783 Z784 Z785 Z786 Z787 Z788 Z789 Z790 Z791 Z792 Z793 Z794 Z795 Z796 Z797 Z798 Z799 Z800 Z801 Z802 Z803 Z804 Z805 Z806 Z807 Z808 Z809 Z810 Z811 Z812 Z813 Z814 Z815 Z816 Z817 Z818 Z819 Z820 Z821 Z822 Z823 Z824 Z825 Z826 Z827 Z828 Z829 Z830 Z831 Z832 Z833 Z834 Z835 Z836 Z837 Z838 Z839 Z840 Z841 Z842 Z843 Z844 Z845 Z846 Z847 Z848 Z849 Z850 Z851 Z852 Z853 Z854 Z855 Z856 Z857 Z858 Z859 Z860 Z861 Z862 Z863 Z864 Z865 Z866 Z867 Z868 Z869 Z870 Z871 Z872 Z873 Z874 Z875 Z876 Z877 Z878 Z879 Z880 Z881 Z882 Z883 Z884 Z885 Z886 Z887 Z888 Z889 Z890 Z891 Z892 Z893 Z894 Z895 Z896 Z897 Z898 Z899 Z900 Z901 Z902 Z903 Z904 Z905 Z906 Z907 Z908 Z909 Z910 Z911 Z912 Z913 Z914 Z915 Z916 Z917 Z918 Z919 Z920 Z921 Z922 Z923 Z924 Z925 Z926 Z927 Z928 Z929 Z930 Z931 Z932 Z933 Z934 Z935 Z936 Z937 Z938 Z939 Z940 Z941 Z942 Z943 Z944 Z945 Z946 Z947 Z948 Z949 Z950 Z951 Z952 Z953 Z954 Z955 Z956 Z957 Z958 Z959 Z960 Z961 Z962 Z963 Z964 Z965 Z966 Z967 Z968 Z969 Z970 Z971 Z972 Z973 Z974 Z975 Z976 Z977 Z978 Z979 Z980 Z981 Z982 Z983 Z984 Z985 Z986 Z987 Z988 Z989 Z990 Z991 Z992 Z993 Z994 Z995 Z996 Z997 Z998 Z999 Z1000 Z1001 Z1002 Z1003 Z1004 Z1005 Z1006 Z1007 Z1008 Z1009 Z1010 Z1011 Z1012 Z1013 Z1014 Z1015 Z1016 Z1017 Z1018 Z1019 Z1020 Z1021 Z1022 Z1023 Z1024 Z1025 Z1026 Z1027 Z1028 Z1029 Z1030 Z1031 Z1032 Z1033 Z1034 Z1035 Z1036 Z1037 Z1038 Z1039 Z1040 Z1041 Z1042 Z1043 Z1044 Z1045 Z1046 Z1047 Z1048 Z1049 Z1050 Z1051 Z1052 Z1053 Z1054 Z1055 Z1056 Z1057 Z1058 Z1059 Z1060 Z1061 Z1062 Z1063 Z1064 Z1065 Z1066 Z1067 Z1068 Z1069 Z1070 Z1071 Z1072 Z1073 Z1074 Z1075 Z1076 Z1077 Z1078 Z1079 Z1080 Z1081 Z1082 Z1083 Z1084 Z1085 Z1086 Z1087 Z1088 Z1089 Z1090 Z1091 Z1092 Z1093 Z1094 Z1095 Z1096 Z1097 Z1098 Z1099 Z1100 Z1101 Z1102 Z1103 Z1104 Z1105 Z1106 Z1107 Z1108 Z1109 Z1110 Z1111 Z1112 Z1113 Z1114 Z1115 Z1116 Z1117 Z1118 Z1119 Z1120 Z1121 Z1122 Z1123 Z1124 Z1125 Z1126 Z1127 Z1128 Z1129 Z1130 Z1131 Z1132 Z1133 Z1134 Z1135 Z1136 Z1137 Z1138 Z1139 Z1140 Z1141 Z1142 Z1143 Z1144 Z1145 Z1146 Z1147 Z1148 Z1149 Z1150 Z1151 Z1152 Z1153 Z1154 Z1155 Z1156 Z1157 Z1158 Z1159 Z1160 Z1161 Z1162 Z1163 Z1164 Z1165 Z1166 Z1167 Z1168 Z1169 Z1170 Z1171 Z1172 Z1173 Z1174 Z1175 Z1176 Z1177 Z1178 Z1179 Z1180 Z1181 Z1182 Z1183 Z1184 Z1185 Z1186 Z1187 Z1188 Z1189 Z1190 Z1191 Z1192 Z1193 Z1194 Z1195 Z1196 Z1197 Z1198 Z1199 Z1200 Z1201 Z1202 Z1203 Z1204 Z1205 Z1206 Z1207 Z1208 Z1209 Z1210 Z1211 Z1212 Z1213 Z1214 Z1215 Z1216 Z1217 Z1218 Z1219 Z1220 Z1221 Z1222 Z1223 Z1224 Z1225 Z1226 Z1227 Z1228 Z1229 Z1230 Z1231 Z1232 Z1233 Z1234 Z1235 Z1236 Z1237 Z1238 Z1239 Z1240 Z1241 Z1242 Z1243 Z1244 Z1245 Z1246 Z1247 Z1248 Z1249 Z1250 Z1251 Z1252 Z1253 Z1254 Z1255 Z1256 Z1257 Z1258 Z1259 Z1260 Z1261 Z1262 Z1263 Z1264 Z1265 Z1266 Z1267 Z1268 Z1269 Z1270 Z1271 Z1272 Z1273 Z1274 Z1275 Z1276 Z1277 Z1278 Z1279 Z1280 Z1281 Z1282 Z1283 Z1284 Z1285 Z1286 Z1287 Z1288 Z1289 Z1290 Z1291 Z1292 Z1293 Z1294 Z1295 Z1296 Z1297 Z1298 Z1299 Z1300 Z1301 Z1302 Z1303 Z1304 Z1305 Z1306 Z1307 Z1308 Z1309 Z1310 Z1311 Z1312 Z1313 Z1314 Z1315 Z1316 Z1317 Z1318 Z1319 Z1320 Z1321 Z1322 Z1323 Z1324 Z1325 Z1326 Z1327 Z1328 Z1329 Z1330 Z1331 Z1332 Z1333 Z1334 Z1335 Z1336 Z1337 Z1338 Z1339 Z1340 Z1341 Z1342 Z1343 Z1344 Z1345 Z1346 Z1347 Z1348 Z1349 Z1350 Z1351 Z1352 Z1353 Z1354 Z1355 Z1356 Z1357 Z1358 Z1359 Z1360 Z1361 Z1362 Z1363 Z1364 Z1365 Z1366 Z1367 Z1368 Z1369 Z1370 Z1371 Z1372 Z1373 Z1374 Z1375 Z1376 Z1377 Z1378 Z1379 Z1380 Z1381 Z1382 Z1383 Z1384 Z1385 Z1386 Z1387 Z1388 Z1389 Z1390 Z1391 Z1392 Z1393 Z1394 Z1395 Z1396 Z1397 Z1398 Z1399 Z1400 Z1401 Z1402 Z1403 Z1404 Z1405 Z1406 Z1407 Z1408 Z1409 Z1410 Z1411 Z1412 Z1413 Z1414 Z1415 Z1416 Z1417 Z1418 Z1419 Z1420 Z1421 Z1422 Z1423 Z1424 Z1425 Z1426 Z1427 Z1428 Z1429 Z1430 Z1431 Z1432 Z1433 Z1434 Z1435 Z1436 Z1437 Z1438 Z1439 Z1440 Z1441 Z1442 Z1443 Z1444 Z1445 Z1446 Z1447 Z1448 Z1449 Z1450 Z1451 Z1452 Z1453 Z1454 Z1455 Z1456 Z1457 Z1458 Z1459 Z1460 Z1461 Z1462 Z1463 Z1464 Z1465 Z1466 Z1467 Z1468 Z1469 Z1470 Z1471 Z1472 Z1473 Z1474 Z1475 Z1476 Z1477 Z1478 Z1479 Z1480 Z1481 Z1482 Z1483 Z1484 Z1485 Z1486 Z1487 Z1488 Z1489 Z1490 Z1491 Z1492 Z1493 Z1494 Z1495 Z1496 Z1497 Z1498 Z1499 Z1500 Z1501 Z1502 Z1503 Z1504 Z1505 Z1506 Z1507 Z1508 Z1509 Z1510 Z1511 Z1512 Z1513 Z1514 Z1515 Z1516 Z1517 Z1518 Z1519 Z1520 Z1521 Z1522 Z1523 Z1524 Z1525 Z1526 Z1527 Z1528 Z1529 Z1530 Z1531 Z1532 Z1533 Z1534 Z1535 Z1536 Z1537 Z1538 Z1539 Z1540 Z1541 Z1542 Z1543 Z1544 Z1545 Z1546 Z1547 Z1548 Z1549 Z1550 Z1551 Z1552 Z1553 Z1554 Z1555 Z1556 Z1557 Z1558 Z1559 Z1560 Z1561 Z1562 Z1563 Z1564 Z1565 Z1566 Z1567 Z1568 Z1569 Z1570 Z1571 Z1572 Z1573 Z1574 Z1575 Z1576 Z1577 Z1578 Z1579 Z1580 Z1581 Z1582 Z1583 Z1584 Z1585 Z1586 Z1587 Z1588 Z1589 Z1590 Z1591 Z1592 Z1593 Z1594 Z1595 Z1596 Z1597 Z1598 Z1599 Z1600 Z1601 Z1602 Z1603 Z1604 Z1605 Z1606 Z1607 Z1608 Z1609 Z1610 Z1611 Z1612 Z1613 Z1614 Z1615 Z1616 Z1617 Z1618 Z1619 Z1620 Z1621 Z1622 Z1623 Z1624 Z1625 Z1626 Z1627 Z1628 Z1629 Z1630 Z1631 Z1632 Z1633 Z1634 Z1635 Z1636 Z1637 Z1638 Z1639 Z1640 Z1641 Z1642 Z1643 Z1644 Z1645 Z1646 Z1647 Z1648 Z1649 Z1650 Z1651 Z1652 Z1653 Z1654 Z1655 Z1656 Z1657 Z1658 Z1659 Z1660 Z1661 Z1662 Z1663 Z1664 Z1665 Z1666 Z1667 Z1668 Z1669 Z1670 Z1671 Z1672 Z1673 Z1674 Z1675 Z1676 Z1677 Z1678 Z1679 Z1680 Z1681 Z1682 Z1683 Z1684 Z1685 Z1686 Z1687 Z1688 Z1689 Z1690 Z1691 Z1692 Z1693 Z1694 Z1695 Z1696 Z1697 Z1698 Z1699 Z1700 Z1701 Z1702 Z1703 Z1704 Z1705 Z1706 Z1707 Z1708 Z1709 Z1710 Z1711 Z1712 Z1713 Z1714 Z1715 Z1716 Z1717 Z1718 Z1719 Z1720 Z1721 Z1722 Z1723 Z1724 Z1725 Z1726 Z1727 Z1728 Z1729 Z1730 Z1731 Z1732 Z1733 Z1734 Z1735 Z1736 Z1737 Z1738 Z1739 Z1740 Z1741 Z1742 Z1743 Z1744 Z1745 Z1746 Z1747 Z1748 Z1749 Z1750 Z1751 Z1752 Z1753 Z1754 Z1755 Z1756 Z1757 Z1758 Z1759 Z1760 Z1761 Z1762 Z1763 Z1764 Z1765 Z1766 Z1767 Z1768 Z1769 Z1770 Z1771 Z1772 Z1773 Z1774 Z1775 Z1776 Z1777 Z1778 Z1779 Z1780 Z1781 Z1782 Z1783 Z1784 Z1785 Z1786 Z1787 Z1788 Z1789 Z1790 Z1791 Z1792 Z1793 Z1794 Z1795 Z1796 Z1797 Z1798 Z1799 Z1800 Z1801 Z1802 Z1803 Z1804 Z1805 Z1806 Z1807 Z1808 Z1809 Z1810 Z1811 Z1812 Z1813 Z1814 Z1815 Z1816 Z1817 Z1818 Z1819 Z1820 Z1821 Z1822 Z1823 Z1824 Z1825 Z1826 Z1827 Z1828 Z1829 Z1830 Z1831 Z1832 Z1833 Z1834 Z1835 Z1836 Z1837 Z1838 Z1839 Z1840 Z1841 Z1842 Z1843 Z1844 Z1845 Z1846 Z1847 Z1848 Z1849 Z1850 Z1851 Z1852 Z1853 Z1854 Z1855 Z1856 Z1857 Z1858 Z1859 Z1860 Z1861 Z1862 Z1863 Z1864 Z1865 Z1866 Z1867 Z1868 Z1869 Z1870 Z1871 Z1872 Z1873 Z1874 Z1875 Z1876 Z1877 Z1878 Z1879 Z1880 Z1881 Z1882 Z1883 Z1884 Z1885 Z1886 Z1887 Z1888 Z1889 Z1890 Z1891 Z1892 Z1893 Z1894 Z1895 Z1896 Z1897 Z1898 Z1899 Z1900 Z1901 Z1902 Z1903 Z1904 Z1905 Z1906 Z1907 Z1908 Z1909 Z1910 Z1911 Z1912 Z1913 Z1914 Z1915 Z1916 Z1917 Z1918 Z1919 Z1920 Z1921 Z1922 Z1923 Z1924 Z1925 Z1926 Z1927 Z1928 Z1929 Z1930 Z1931 Z1932 Z1933 Z1934 Z1935 Z1936 Z1937 Z1938 Z1939 Z1940 Z1941 Z1942 Z1943 Z1944 Z1945 Z1946 Z1947 Z1948 Z1949 Z1950 Z1951 Z1952 Z1953 Z1954 Z1955 Z1956 Z1957 Z1958 Z1959 Z1960 Z1961 Z1962 Z1963 Z1964 Z1965 Z1966 Z1967 Z1968 Z1969 Z1970 Z1971 Z1972 Z1973 Z1974 Z1975 Z1976 Z1977 Z1978 Z1979 Z1980 Z1981 Z1982 Z1983 Z1984 Z1985 Z1986 Z1987 Z1988 Z1989 Z1990 Z1991 Z1992 Z1993 Z1994 Z1995 Z1996 Z1997 Z1998 Z1999 Z2000 Z2001 Z2002 Z2003 Z2004 Z2005 Z2006 Z2007 Z2008 Z2009 Z2010 Z2011 Z2012 Z2013 Z2014 Z2015 Z2016 Z2017 Z2018 Z2019 Z2020 Z2021 Z2022 Z2023 Z2024 Z2025 Z2026 Z2027 Z2028 Z2029 Z2030 Z2031 Z2032 Z2033 Z2034 Z2035 Z2036 Z2037 Z2038 Z2039 Z2040 Z2041 Z2042 Z2043 Z2044 Z2045 Z2046 Z2047 Z2048 Z2049 Z2050 Z2051 Z2052 Z2053 Z2054 Z2055 Z2056 Z2057 Z2058 Z2059 Z2060 Z2061 Z2062 Z2063 Z2064 Z2065 Z2066 Z2067 Z2068 Z2069 Z2070 Z2071 Z2072 Z2073 Z2074 Z2075 Z2076 Z2077 Z2078 Z2079 Z2080 Z2081 Z2082 Z2083 Z2084 Z2085 Z2086 Z2087 Z2088 Z2089 Z2090 Z2091 Z2092 Z2093 Z2094 Z2095 Z2096 Z2097 Z2098 Z2099 Z2100 Z2101 Z2102 Z2103 Z2104 Z2105 Z2106 Z2107 Z2108 Z2109 Z2110 Z2111 Z2112 Z2113 Z2114 Z2115 Z2116 Z2117 Z2118 Z2119 Z2120 Z2121 Z2122 Z2123 Z2124 Z2125 Z2126 Z2127 Z2128 Z2129 Z2130 Z2131 Z2132 Z2133 Z2134 Z2135 Z2136 Z2137 Z2138 Z2139 Z2140 Z2141 Z2142 Z2143 Z2144 Z2145 Z2146 Z2147 Z2148 Z2149 Z2150 Z2151 Z2152 Z2153 Z2154 Z2155 Z2156 Z2157 Z2158 Z2159 Z2160 Z2161 Z2162 Z2163 Z2164 Z2165 Z2166 Z2167 Z2168 Z2169 Z2170 Z2171 Z2172 Z2173 Z2174 Z2175 Z2176 Z2177 Z2178 Z2179 Z2180 Z2181 Z2182 Z2183 Z2184 Z2185 Z2186 Z2187 Z2188 Z2189 Z2190 Z2191 Z2192 Z2193 Z2194 Z2195 Z2196 Z2197 Z2198 Z2199 Z2200 Z2201 Z2202 Z2203 Z2204 Z2205 Z2206 Z2207 Z2208 Z2209 Z2210 Z2211 Z2212 Z2213 Z2214 Z2215 Z2216 Z2217 Z2218 Z2219 Z2220 Z2221 Z2222 Z2223 Z2224 Z2225 Z2226 Z2227 Z2228 Z2229 Z2230 Z2231 Z2232 Z2233 Z2234 Z2235 Z2236 Z2237 Z2238 Z2239 Z2240 Z2241 Z2242 Z2243 Z2244 Z2245 Z2246 Z2247 Z2248 Z2249 Z2250 Z2251 Z2252 Z2253 Z2254 Z2255 Z2256 Z2257 Z2258 Z2259 Z2260 Z2261 Z2262 Z2263 Z2264 Z2265 Z2266 Z2267 Z2268 Z2269 Z2270 Z2271 Z2272 Z2273 Z2274 Z2275 Z22
```