

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

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

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 Fifth Annual Conference of the Southeast SAS Users Group*, 5.

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

```

/* *****
* MACRO: COMPARE NOTE: uses macro ARRAY, NOBS
*
* USAGE: 1. %COMPARE(DATA1,DAT2,IDX);
* 2. %COMPARE(DATA1,DAT2,IDX,PRINTFILE=filename);
* 3. %COMPARE(DATA1,DAT2,IDX,TESTING=1);
* 4. %COMPARE(DATA1,DAT2,IDX,DROPLIST=Var1 Var2 .. VarN);
* recommended drop character variables whose length gt 12
* 5. %COMPARE(DATA1,DAT2,IDX,READBASE=XXXX,READCOMP=YYYY);
* 6. TITLE1 'title for this project';
* %COMPARE(DATA1,DAT2,IDX,TITLE=2);
* 7. %COMPARE(DATA1,DAT2,IDX,IDX2,SDM_IDX=2);
*
* DESCRIPTION: use proc COMPARE output data set
* to write corex file
* and summary report of differences
*
* PROCES: 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. var is 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, S.S. C.Sc 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
IDXLIST /* list of primary-key(s) == by-vars

```

```

SDM_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 /* feasible test msg and prints?
TITLE1 = /* TITLE line 1
)/* *****
TITLE2=TITLE. "comparison of BASE. and COMPARE."

%*1. if IDXLIST blank, print err-msg, exit;
%IF &IDXLIST = %THEN %DO; %put *****;
%put !! no ID-list !!;
%put *****; %NOTO ENDOMACR; %END;

%*2. %NOBS of BASE & COMPARE for error-check;
%local BASENOBS; %NOBS(BASENOBS,data = BASE );
%local COMPNOBS; %NOBS(COMPNOBS,data = COMPARE);

%*3. if err-checks print err-msg, exit;
%IF BASENOBS ne COMPNOBS /*-----
or not BASENOBS
or not COMPNOBS %THEN %DO; DATA _NULL_;
title SPRTFILE;
put
%IF BASENOBS ne COMPNOBS %THEN *****number of obs differ, "
BASE. = BASENOBS. "
COMPARE. = COMPNOBS. "
/*
%IF not BASENOBS %THEN
*****no obs in BASE.;
%IF not COMPNOBS %THEN
*****no obs in COMPARE.;
/*part closure*/
%NOTO ENDOMACR; /*-----*/ %END;

%*4. %ARRAY of ID(s): for use in macro loops;
%local I; %DO I = 1 %TO &SDM_IDX; %local IDXEX.; %END;
%LET IDXLIST = %upcase(&IDXLIST.); /* proc CONTENTS name is upcase */
%ARRAY(IDX,&IDXLIST);

%*5. var is IDXLIST (character? numeric?);
proc CONTENTS data = BASE.
%IF 'READBASE' ne "." %THEN (read=READBASE.);
out = BASECORT (keep = Base Type) noprint;

```

```

DATA _NULL_;
do until(EndOfFile);
  set BASECONT 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_ID;
  if Name = 'SAIDSEL.' then call symput('TYPEL',C_N); *END;
  /*-----*/ do until(EndOfFile); *END;
stop;

%*6. proc COMPARE
base = BASE. (%IF 'SDROPLIST' ne '' %THEN drop=SDROPLIST.;
%IF 'SREADBASE' ne '' %THEN read=SREADBASE.;)
compare = %COMPARE(%IF 'SDROPLIST' ne '' %THEN drop=SDROPLIST.;
%IF 'SREADCOMP' ne '' %THEN read=SREADCOMP.;)
out = COMPARE;
outall /* = OUTBASE OUTCOMP OUTDIF OUTPERCENT */
outroqual /* write only obs with one or more differences */
noprnt;
by SIDLIST.;

%macro TESTPRINT(DATA,MSG);/*-----*/
%IF %TESTING %THEN %DO; proc PRINT data=DATA;
  TITLE%eval(&ITITLE.+1) "data=DATA; MSG";
  proc CONTENTS data = DATA;
  /*-----*/ run; TITLE%eval(&ITITLE.+1);/*-----*/%END; %END;
%TESTPRINT(COMPARE,output from proc COMPARE);

%*7. %MOSS, if no differences print exit-msg, exit;
%local ANYDIF; %MOSS(ANYDIF);

%IF not ANYDIF %THEN %DO; DATA _NULL_;
  file SPRTFILE;
  put '***no differences;';
  %&OTO ENDMACR; %END;

%*8. TRANPOSE: rotate & unique data to get one obs with one difference;
proc TRANSPOSE data = COMPARE (drop = _obs_)
  out = COMPARE (drop = _label_);
  id _TYPE_;
  by SIDLIST.;
  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 SID1;
  %IF SDIN_ID; go 2 %THEN /*do cross-tab of IDs */
  %DO I = 2 %TO SDIN_ID; * SAIDSEL. %END;
  / list noprnt out = DIFFIDS;
  tables _Name_ / list noprnt out = DIFFVAR;

%local DIFFIDS; %MOSS(DIFFIDS,data = DIFFIDS);
%local DIFFVAR; %MOSS(DIFFVAR,data = DIFFVAR);
%local DIFFCLS; %MOSS(DIFFCLS,data = COMPARE);
%local BASEVAR; %MOSS(BASEVAR,data = BASECONT);

%*10. write corex to PRNTFILE;
DATA %IF 'SPRTFILE' eq 'PRINT' %THEN _NULL_;
  %ELSE COMPARE; /*DATA closure*/;
  options pageno=1;
  drop S & C1 C2 C3 DIF Percent HDIFF;
  length S & 1;
  retain S -1 HDIFF 0 S 1 /*counter for proc PRINT sum*/;
  file SPRTFILE;
  do until(EndOfFile);
    set COMPARE end = EndOfFile;
    by SIDLIST;

    if DIF ne Percent then S = ' '; /* = var is numeric */
    else S = ' '; /* = var is char */

    %MACRO Q(N);/*returns quote for character variables;
      parm is reference */
    %IF %STR(%N) = C %THEN ' ';/*-----*/%END;

    if first.SAIDSEL_ID then do; HDIFF = 0; /*-----*/
      put ' ' if SID1 = ' ' %R(1) SID1 -S %R(1)
      %IF SDIN_ID; go 2 %THEN
      %DO I = 2 %TO SDIN_ID; * and SAIDSEL = ' ' %R(1) SAIDSEL -S %R(1)
      %END;
      % then do; /*-----*/%IF first.%/* END;

    if Base ne Compare then do; HDIFF + 1; /*-----*/
      put ' ' _Name_ SID1 ' ' S1 S & S Base -S S ' '
      / ' ' _Name_ SID1 ' ' S1 S & S Compare -S S ' ' ;

```

```

  if length(SID1) gt 5
  then put ' ' diffperc SID1 ' ' S1 S & S DIF -S S ' ' ;
  /*----if Base ne Compare */ end;

  if last.SAIDSEL_ID then put ' ' end;HDIFF = HDIFF ' ';
  %IF 'SPRTFILE' ne 'PRINT' %THEN %DO; output; %END;
  /*-----*/ do until(EndOfFile); *END;
  %macro MAKEPCHT(VAR,NUM,DECM);/*create %, trim value to SD decimals -*/
  %local D I TRIMLEN; %LET D = 3; %LET TRIMLEN = 0;
  %LET VAR = %sysvalf(100 * %SDNUM / SAIDSEL);
  %LET I = %index(SAIDSEL.);/*if dot in value trim number of decimals;
  %IF %I %THEN %DO; %LET TRIMLEN = %eval(%length(SAIDSEL) - %I);
  %LET TRIMLEN gt SD %THEN %LET TRIMLEN = SD;
  %LET VAR = %substr(SAIDSEL,1,%I + TRIMLEN); %END;
  %PUT VAR = SAIDSEL; /*-----*/%END;
  %local BASECLS; %LET BASECLS = %eval(%BASEVAR * SAIDSEL);
  %local PCHTHOSS; %MAKEPCHT(PCHTHOSS,DIFFIDS, _BASEVAR);
  %local PCHTVARS; %MAKEPCHT(PCHTVARS,DIFFVAR, _BASEVAR);
  %local PCHTCLS; %MAKEPCHT(PCHTCLS,DIFFCLS, _BASECLS);
  retain C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31 C32 C33 C34 C35 C36 C37 C38 C39 C40 C41 C42 C43 C44 C45 C46 C47 C48 C49 C50 C51 C52 C53 C54 C55 C56 C57 C58 C59 C60 C61 C62 C63 C64 C65 C66 C67 C68 C69 C70 C71 C72 C73 C74 C75 C76 C77 C78 C79 C80 C81 C82 C83 C84 C85 C86 C87 C88 C89 C90 C91 C92 C93 C94 C95 C96 C97 C98 C99 C100 C101 C102 C103 C104 C105 C106 C107 C108 C109 C110 C111 C112 C113 C114 C115 C116 C117 C118 C119 C120 C121 C122 C123 C124 C125 C126 C127 C128 C129 C130 C131 C132 C133 C134 C135 C136 C137 C138 C139 C140 C141 C142 C143 C144 C145 C146 C147 C148 C149 C150 C151 C152 C153 C154 C155 C156 C157 C158 C159 C160 C161 C162 C163 C164 C165 C166 C167 C168 C169 C170 C171 C172 C173 C174 C175 C176 C177 C178 C179 C180 C181 C182 C183 C184 C185 C186 C187 C188 C189 C190 C191 C192 C193 C194 C195 C196 C197 C198 C199 C200 C201 C202 C203 C204 C205 C206 C207 C208 C209 C210 C211 C212 C213 C214 C215 C216 C217 C218 C219 C220 C221 C222 C223 C224 C225 C226 C227 C228 C229 C230 C231 C232 C233 C234 C235 C236 C237 C238 C239 C240 C241 C242 C243 C244 C245 C246 C247 C248 C249 C250 C251 C252 C253 C254 C255 C256 C257 C258 C259 C260 C261 C262 C263 C264 C265 C266 C267 C268 C269 C270 C271 C272 C273 C274 C275 C276 C277 C278 C279 C280 C281 C282 C283 C284 C285 C286 C287 C288 C289 C290 C291 C292 C293 C294 C295 C296 C297 C298 C299 C300 C301 C302 C303 C304 C305 C306 C307 C308 C309 C310 C311 C312 C313 C314 C315 C316 C317 C318 C319 C320 C321 C322 C323 C324 C325 C326 C327 C328 C329 C330 C331 C332 C333 C334 C335 C336 C337 C338 C339 C340 C341 C342 C343 C344 C345 C346 C347 C348 C349 C350 C351 C352 C353 C354 C355 C356 C357 C358 C359 C360 C361 C362 C363 C364 C365 C366 C367 C368 C369 C370 C371 C372 C373 C374 C375 C376 C377 C378 C379 C380 C381 C382 C383 C384 C385 C386 C387 C388 C389 C390 C391 C392 C393 C394 C395 C396 C397 C398 C399 C400 C401 C402 C403 C404 C405 C406 C407 C408 C409 C410 C411 C412 C413 C414 C415 C416 C417 C418 C419 C420 C421 C422 C423 C424 C425 C426 C427 C428 C429 C430 C431 C432 C433 C434 C435 C436 C437 C438 C439 C440 C441 C442 C443 C444 C445 C446 C447 C448 C449 C450 C451 C452 C453 C454 C455 C456 C457 C458 C459 C460 C461 C462 C463 C464 C465 C466 C467 C468 C469 C470 C471 C472 C473 C474 C475 C476 C477 C478 C479 C480 C481 C482 C483 C484 C485 C486 C487 C488 C489 C490 C491 C492 C493 C494 C495 C496 C497 C498 C499 C500 C501 C502 C503 C504 C505 C506 C507 C508 C509 C510 C511 C512 C513 C514 C515 C516 C517 C518 C519 C520 C521 C522 C523 C524 C525 C526 C527 C528 C529 C530 C531 C532 C533 C534 C535 C536 C537 C538 C539 C540 C541 C542 C543 C544 C545 C546 C547 C548 C549 C550 C551 C552 C553 C554 C555 C556 C557 C558 C559 C560 C561 C562 C563 C564 C565 C566 C567 C568 C569 C570 C571 C572 C573 C574 C575 C576 C577 C578 C579 C580 C581 C582 C583 C584 C585 C586 C587 C588 C589 C590 C591 C592 C593 C594 C595 C596 C597 C598 C599 C600 C601 C602 C603 C604 C605 C606 C607 C608 C609 C610 C611 C612 C613 C614 C615 C616 C617 C618 C619 C620 C621 C622 C623 C624 C625 C626 C627 C628 C629 C630 C631 C632 C633 C634 C635 C636 C637 C638 C639 C640 C641 C642 C643 C644 C645 C646 C647 C648 C649 C650 C651 C652 C653 C654 C655 C656 C657 C658 C659 C660 C661 C662 C663 C664 C665 C666 C667 C668 C669 C670 C671 C672 C673 C674 C675 C676 C677 C678 C679 C680 C681 C682 C683 C684 C685 C686 C687 C688 C689 C690 C691 C692 C693 C694 C695 C696 C697 C698 C699 C700 C701 C702 C703 C704 C705 C706 C707 C708 C709 C710 C711 C712 C713 C714 C715 C716 C717 C718 C719 C720 C721 C722 C723 C724 C725 C726 C727 C728 C729 C730 C731 C732 C733 C734 C735 C736 C737 C738 C739 C740 C741 C742 C743 C744 C745 C746 C747 C748 C749 C750 C751 C752 C753 C754 C755 C756 C757 C758 C759 C760 C761 C762 C763 C764 C765 C766 C767 C768 C769 C770 C771 C772 C773 C774 C775 C776 C777 C778 C779 C780 C781 C782 C783 C784 C785 C786 C787 C788 C789 C790 C791 C792 C793 C794 C795 C796 C797 C798 C799 C800 C801 C802 C803 C804 C805 C806 C807 C808 C809 C810 C811 C812 C813 C814 C815 C816 C817 C818 C819 C820 C821 C822 C823 C824 C825 C826 C827 C828 C829 C830 C831 C832 C833 C834 C835 C836 C837 C838 C839 C840 C841 C842 C843 C844 C845 C846 C847 C848 C849 C850 C851 C852 C853 C854 C855 C856 C857 C858 C859 C860 C861 C862 C863 C864 C865 C866 C867 C868 C869 C870 C871 C872 C873 C874 C875 C876 C877 C878 C879 C880 C881 C882 C883 C884 C885 C886 C887 C888 C889 C890 C891 C892 C893 C894 C895 C896 C897 C898 C899 C900 C901 C902 C903 C904 C905 C906 C907 C908 C909 C910 C911 C912 C913 C914 C915 C916 C917 C918 C919 C920 C921 C922 C923 C924 C925 C926 C927 C928 C929 C930 C931 C932 C933 C934 C935 C936 C937 C938 C939 C940 C941 C942 C943 C944 C945 C946 C947 C948 C949 C950 C951 C952 C953 C954 C955 C956 C957 C958 C959 C960 C961 C962 C963 C964 C965 C966 C967 C968 C969 C970 C971 C972 C973 C974 C975 C976 C977 C978 C979 C980 C981 C982 C983 C984 C985 C986 C987 C988 C989 C990 C991 C992 C993 C994 C995 C996 C997 C998 C999 C1000 C1001 C1002 C1003 C1004 C1005 C1006 C1007 C1008 C1009 C1010 C1011 C1012 C1013 C1014 C1015 C1016 C1017 C1018 C1019 C1020 C1021 C1022 C1023 C1024 C1025 C1026 C1027 C1028 C1029 C1030 C1031 C1032 C1033 C1034 C1035 C1036 C1037 C1038 C1039 C1040 C1041 C1042 C1043 C1044 C1045 C1046 C1047 C1048 C1049 C1050 C1051 C1052 C1053 C1054 C1055 C1056 C1057 C1058 C1059 C1060 C1061 C1062 C1063 C1064 C1065 C1066 C1067 C1068 C1069 C1070 C1071 C1072 C1073 C1074 C1075 C1076 C1077 C1078 C1079 C1080 C1081 C1082 C1083 C1084 C1085 C1086 C1087 C1088 C1089 C1090 C1091 C1092 C1093 C1094 C1095 C1096 C1097 C1098 C1099 C1100 C1101 C1102 C1103 C1104 C1105 C1106 C1107 C1108 C1109 C1110 C1111 C1112 C1113 C1114 C1115 C1116 C1117 C1118 C1119 C1120 C1121 C1122 C1123 C1124 C1125 C1126 C1127 C1128 C1129 C1130 C1131 C1132 C1133 C1134 C1135 C1136 C1137 C1138 C1139 C1140 C1141 C1142 C1143 C1144 C1145 C1146 C1147 C1148 C1149 C1150 C1151 C1152 C1153 C1154 C1155 C1156 C1157 C1158 C1159 C1160 C1161 C1162 C1163 C1164 C1165 C1166 C1167 C1168 C1169 C1170 C1171 C1172 C1173 C1174 C1175 C1176 C1177 C1178 C1179 C1180 C1181 C1182 C1183 C1184 C1185 C1186 C1187 C1188 C1189 C1190 C1191 C1192 C1193 C1194 C1195 C1196 C1197 C1198 C1199 C1200 C1201 C1202 C1203 C1204 C1205 C1206 C1207 C1208 C1209 C1210 C1211 C1212 C1213 C1214 C1215 C1216 C1217 C1218 C1219 C1220 C1221 C1222 C1223 C1224 C1225 C1226 C1227 C1228 C1229 C1230 C1231 C1232 C1233 C1234 C1235 C1236 C1237 C1238 C1239 C1240 C1241 C1242 C1243 C1244 C1245 C1246 C1247 C1248 C1249 C1250 C1251 C1252 C1253 C1254 C1255 C1256 C1257 C1258 C1259 C1260 C1261 C1262 C1263 C1264 C1265 C1266 C1267 C1268 C1269 C1270 C1271 C1272 C1273 C1274 C1275 C1276 C1277 C1278 C1279 C1280 C1281 C1282 C1283 C1284 C1285 C1286 C1287 C1288 C1289 C1290 C1291 C1292 C1293 C1294 C1295 C1296 C1297 C1298 C1299 C1300 C1301 C1302 C1303 C1304 C1305 C1306 C1307 C1308 C1309 C1310 C1311 C1312 C1313 C1314 C1315 C1316 C1317 C1318 C1319 C1320 C1321 C1322 C1323 C1324 C1325 C1326 C1327 C1328 C1329 C1330 C1331 C1332 C1333 C1334 C1335 C1336 C1337 C1338 C1339 C1340 C1341 C1342 C1343 C1344 C1345 C1346 C1347 C1348 C1349 C1350 C1351 C1352 C1353 C1354 C1355 C1356 C1357 C1358 C1359 C1360 C1361 C1362 C1363 C1364 C1365 C1366 C1367 C1368 C1369 C1370 C1371 C1372 C1373 C1374 C1375 C1376 C1377 C1378 C1379 C1380 C1381 C1382 C1383 C1384 C1385 C1386 C1387 C1388 C1389 C1390 C1391 C1392 C1393 C1394 C1395 C1396 C1397 C1398 C1399 C1400 C1401 C1402 C1403 C1404 C1405 C1406 C1407 C1408 C1409 C1410 C1411 C1412 C1413 C1414 C1415 C1416 C1417 C1418 C1419 C1420 C1421 C1422 C1423 C1424 C1425 C1426 C1427 C1428 C1429 C1430 C1431 C1432 C1433 C1434 C1435 C1436 C1437 C1438 C1439 C1440 C1441 C1442 C1443 C1444 C1445 C1446 C1447 C1448 C1449 C1450 C1451 C1452 C1453 C1454 C1455 C1456 C1457 C1458 C1459 C1460 C1461 C1462 C1463 C1464 C1465 C1466 C1467 C1468 C1469 C1470 C1471 C1472 C1473 C1474 C1475 C1476 C1477 C1478 C1479 C1480 C1481 C1482 C1483 C1484 C1485 C1486 C1487 C1488 C1489 C1490 C1491 C1492 C1493 C1494 C1495 C1496 C1497 C1498 C1499 C1500 C1501 C1502 C1503 C1504 C1505 C1506 C1507 C1508 C1509 C1510 C1511 C1512 C1513 C1514 C1515 C1516 C1517 C1518 C1519 C1520 C1521 C1522 C1523 C1524 C1525 C1526 C1527 C1528 C1529 C1530 C1531 C1532 C1533 C1534 C1535 C1536 C1537 C1538 C1539 C1540 C1541 C1542 C1543 C1544 C1545 C1546 C1547 C1548 C1549 C1550 C1551 C1552 C1553 C1554 C1555 C1556 C1557 C1558 C1559 C1560 C1561 C1562 C1563 C1564 C1565 C1566 C1567 C1568 C1569 C1570 C1571 C1572 C1573 C1574 C1575 C1576 C1577 C1578 C1579 C1580 C1581 C1582 C1583 C1584 C1585 C1586 C1587 C1588 C1589 C1590 C1591 C1592 C1593 C1594 C1595 C1596 C1597 C1598 C1599 C1600 C1601 C1602 C1603 C1604 C1605 C1606 C1607 C1608 C1609 C1610 C1611 C1612 C1613 C1614 C1615 C1616 C1617 C1618 C1619 C1620 C1621 C1622 C1623 C1624 C1625 C1626 C1627 C1628 C1629 C1630 C1631 C1632 C1633 C1634 C1635 C1636 C1637 C1638 C1639 C1640 C1641 C1642 C1643 C1644 C1645 C1646 C1647 C1648 C1649 C1650 C1651 C1652 C1653 C1654 C1655 C1656 C1657 C1658 C1659 C1660 C1661 C1662 C1663 C1664 C1665 C1666 C1667 C1668 C1669 C1670 C1671 C1672 C1673 C1674 C1675 C1676 C1677 C1678 C1679 C1680 C1681 C1682 C1683 C1684 C1685 C1686 C1687 C1688 C1689 C1690 C1691 C1692 C1693 C1694 C1695 C1696 C1697 C1698 C1699 C1700 C1701 C1702 C1703 C1704 C1705 C1706 C1707 C1708 C1709 C1710 C1711 C1712 C1713 C1714 C1715 C1716 C1717 C1718 C1719 C1720 C1721 C1722 C1723 C1724 C1725 C1726 C1727 C1728 C1729 C1730 C1731 C1732 C1733 C1734 C1735 C1736 C1737 C1738 C1739 C1740 C1741 C1742 C1743 C1744 C1745 C1746 C1747 C1748 C1749 C1750 C1751 C1752 C1753 C1754 C1755 C1756 C1757 C1758 C1759 C1760 C1761 C1762 C1763 C1764 C1765 C1766 C1767 C1768 C1769 C1770 C1771 C1772 C1773 C1774 C1775 C1776 C1777 C1778 C1779 C1780 C1781 C1782 C1783 C1784 C1785 C1786 C1787 C1788 C1789 C1790 C1791 C1792 C1793 C1794 C1795 C1796 C1797 C1798 C1799 C1800 C1801 C1802 C1803 C1804 C1805 C1806 C1807 C1808 C1809 C1810 C1811 C1812 C1813 C1814 C1815 C1816 C1817 C1818 C1819 C1820 C1821 C1822 C1823 C1824 C1825 C1826 C1827 C1828 C1829 C1830 C1831 C1832 C1833 C1834 C1835 C1836 C1837 C1838 C1839 C1840 C1841 C1842 C1843 C1844 C1845 C1846 C1847 C1848 C1849 C1850 C1851 C1852 C1853 C1854 C1855 C1856 C1857 C1858 C1859 C1860 C1861 C1862 C1863 C1864 C1865 C1866 C1867 C1868 C1869 C1870 C1871 C1872 C1873 C1874 C1875 C1876 C1877 C1878 C1879 C1880 C1881 C1882 C1883 C1884 C1885 C1886 C1887 C1888 C1889 C1890 C1891 C1892 C1893 C1894 C1895 C1896 C1897 C1898 C1899 C1900 C1901 C1902 C1903 C1904 C1905 C1906 C1907 C1908 C1909 C1910 C1911 C1912 C1913 C1914 C1915 C1916 C1917 C1918 C1919 C1920 C1921 C1922 C1923 C1924 C1925 C1926 C1927 C1928 C1929 C1930 C1931 C1932 C1933 C1934 C1935 C1936 C1937 C1938 C1939 C1940 C1941 C1942 C1943 C1944 C1945 C1946 C1947 C1948 C1949 C1950 C1951 C1952 C1953 C1954 C1955 C1956 C1957 C1958 C1959 C1960 C1961 C1962 C1963 C1964 C1965 C1966 C1967 C1968 C1969 C1970 C1971 C1972 C1973 C1974 C1975 C1976 C1977 C1978 C1979 C1980 C1981 C1982 C1983 C1984 C1985 C1986 C1987 C1988 C1989 C1990 C1991 C1992 C1993 C1994 C1995 C1996 C1997 C1998 C1999 C2000 C2001 C2002 C2003 C2004 C2005 C2006 C2007 C2008 C2009 C2010 C2011 C2012 C2013 C2014 C2015 C2016 C2017 C2018 C2019 C2020 C2021 C2022 C2023 C2024 C2025 C2026 C2027 C2028 C2029 C2030 C2031 C2032 C2033 C2034 C2035 C2036 C2037 C2038 C2039 C2040 C2041 C2042 C2043 C2044 C2045 C2046 C2047 C2048 C2049 C2050 C2051 C2052 C2053 C2054 C2055 C2056 C2057 C2058 C2059 C2060 C2061 C2062 C2063 C2064 C2065 C2066 C2067 C2068 C2069 C2070 C2071 C2072 C2073 C2074 C2075 C2076 C2077 C2078 C2079 C2080 C2081 C2082 C2083 C2084 C2085 C2086 C2087 C2088 C2089 C2090 C2091 C2092 C2093 C2094 C2095 C2096 C2097 C2098 C2099 C2100 C2101 C2102 C2103 C2104 C2105 C2106 C2107 C2108 C2109 C2110 C2111 C2112 C2113 C2114 C2115 C2116 C2117 C2118 C2119 C2120 C2121 C2122 C2123 C2124 C2125 C2126 C2127 C2128 C2129 C2130 C2131 C2132 C2133 C2134 C2135 C2136 C2137 C2138 C2139 C2140 C2141 C2142 C2143 C2144 C2145 C2146 C2147 C2148 C2149 C2150 C2151 C2152 C2153 C2154 C2155 C2156 C2157 C2158 C2159 C2160 C2161 C2162 C2163 C2164 C2165 C2166 C2167 C2168 C2169 C2170 C2171 C2172 C2173 C2174 C2175 C2176 C2177 C2178 C2179 C2180 C2181 C2182 C2183 C2184 C2185 C2186 C2187 C2188 C2189 C2190 C2191 C2192 C2193 C2194 C2195 C2196 C2197 C2198 C2199 C2200 C2201 C2202 C2203 C2204 C2205 C2206 C2207 C2208 C2209 C2210 C2211 C2212 C2213 C2214 C2215 C2216 C2217 C2218 C2219 C2220 C2221 C2222 C2223 C2224 C2225 C2226 C2227 C2228 C2229 C2230 C2231 C2232 C2233 C2234 C2235 C2236 C2237 C2238 C2239 C2240 C2241 C2242 C2243 C2244 C2245 C2246 C2247 C2248 C2249 C2250 C2251 C2252 C2253 C2254 C2255 C2256 C2257 C2258 C2259 C2260 C2261 C2262 C2263 C2264 C2265 C2266 C2267 C2268 C2269 C2270 C2271 C2272 C2273 C2274 C2275 C2276 C2277 C2278 C2279 C2280 C2281 C2282 C2283 C2284 C2285 C2286 C2287 C2288 C2289 C2290 C2291 C2292 C2293 C2294 C2295 C2296 C2297 C2298 C2299 C2300 C2301 C2302 C2303 C2304 C2305 C2306 C2307 C2308 C2309 C2310 C2311 C2312 C2313 C2314 C2315 C2316 C2317 C2318 C2319 C2320 C2321 C2322 C2323 C2324 C2325 C2326 C2327 C2328 C2329 C2330 C2331 C2332 C2333 C2334 C2335 C2336 C2337 C2338 C2339 C2340 C2341 C2342 C2343 C2344 C2345 C2346 C2347 C2348 C2349 C2350 C2351 C2352 C2353 C2354 C2355 C2356 C2357 C2358 C2359 C2360 C2361 C2362 C2363 C2364 C2365 C2366 C2367 C2368 C2369 C237
```