The researcher frequently needs to replace the values of some or all variables by new values. Currently, the only way this operation can be performed within the Statistical Analysis System is through use of multiple IF-THEN statements. This approach is often lengthy and inefficient as well as being inadequate for some types of value changes. RECODE has been designed and is being developed as a comprehensive and efficient solution to the problem of value substitution.

The RECODE procedure features a simple, SAS-like syntax and incorporates typical SAS parameters and options such as the OUTPUT statement and a FUZZ parameter. Additionally, RECODE is able to handle a wide variety of recoding situations, such as mapping a range of values into a single value, converting numeric values to character values and character values to numeric values, searching for high and low values of a variable, and performing limited character string manipulation. The usual SAS lists are supported, as well as the special variable names ALL, _ALLALPHA_, and _ALLNUM_. An EXCEPT list is also available for use with variable lists to save coding time.

The sole function of RECODE is to create a SAS data set containing the specified value substitutions. The essence of how to use RECODE is shown in the example which follows:

```sas
DATA; INPUT AGE @@; CARDS;
10 14 42 49 32 11 53 26 99 . 21
PROC RECODE;
RECODE AGE 0 THRU 19=1 20 THRU 29=2
30 THRU 39=3 40 THRU 49=4 50 THRU 59=5
99=.;
```

This example demonstrates mapping a numeric value or range into another numeric value. Each age is changed to an age category, the value 99 is changed to a missing value, and the missing value in the original data set is left unchanged. More complicated mappings can be handled by RECODE with similar ease.

The general form of the procedure statements for RECODE is as follows:

```sas
PROC RECODE procedure options and parameters;
RECODE variables specification
value change specifications;
recode options and parameters;
```

The procedure options and parameters which may be used are:

- `DATA =` data set name specifies the name of the input data set.
- `OUT =` data set name specifies the name of the output data set.

The following examples are legitimate statements to initiate a RECODE procedure.

```sas
PROC RECODE;
PROC RECODE DATA=IN.TOTALS2;
PROC RECODE F=.5 OUT=EDITED;
```

The RECODE statements give specific information concerning the variables which are to be changed and the substitutions which are to take place. Any number of RECODE statements may be used in a single RECODE procedure to accomplish the desired changes to the input data set. The variables specification portion of the RECODE statement may include the usual SAS lists as well as ALL, ALLNUM, and _ALLALPHA_ to specify, respectively, that all variables, all numeric variables, or all alphanumeric variables are to be used in the recode procedure. An EXCEPT list is supported for use in conjunction with a variables specification to shorten coding time when dealing with a large number of variables. Variable lists which are permissible include the following:

- `RECODE SEX RACE recode specifications;`
- `RECODE AGE TEMP Q1-Q832 (EXCEPT Q238 Q423) recode specifications;`
- `RECODE _ALL_ recode specifications;`
- `RECODE _ALLALPHA_ (EXCEPT RACE) recode specifications;`

A value change specification usually follows the variables specification in the RECODE statement. The specification includes the current value, either implied or specifically designated, to the left of an equals sign and the value to which it should be mapped to the right of the equals. The keywords HI, LO, and THRU are used to the left of the equals when referring to a range of values. HI and LO refer to the highest and lowest values which occur for the variable under con-
Consideration. Alphanumeric as well as numeric values may be recoded, but alphanumerics must appear in single quotes both to the left and right of the equals sign. The following are legitimate value change specifications:

```
9=1 8=2 7=3 LO THRU 6=4
'W'=1 'B'=2 'O'=3 'E'=
1='FEMALE' 2='MALE',
LO THRU 'ALBERT'=1 'ZEKR' THRU HI=99
```

The discussion up to this point has centered around mapping specific values or a range of values to a particular destination. RECODE also has the capability to search character strings for all values which begin with a specified set of n bytes. For example, the specification 'A': = 'BC': =2 maps all character values beginning with an A to the value 1 and those beginning with BC to the value 2. A character value of length n followed by a colon may be used to replace the first n bytes of a character value. For example, the specification 'AL': = 'ALBERT': would cause all values beginning with AL in the input data set to begin with ALBERT in the output data set.

The RECODE options and parameters are CONVERT and ELSE. They tell RECODE how to handle values which have not been mapped by previous specifications.

CONVERT indicates how a variable type change should be handled and applies only to values which have not been mapped by previous specifications within the current RECODE statement. When making a character to numeric mapping, the keyword CONVERT causes character values such as '0' and '233' to be mapped to their numeric counterparts. In a numeric to character mapping procedure, CONVERT causes all numerics to become their character counterparts. That is, 123 becomes '123', 67 becomes '67' and . becomes '.

ELSE indicates the mapping destination for values not mapped by a specific instruction or CONVERT. ELSE=99 maps all unmapped values to the numeric value 99.

RECODE can be a vital tool in a large variety of applications. The following examples demonstrate some of the more readily apparent situations in which RECODE could be a highly effective data manipulation tool.

Example 1: Data tapes are frequently compiled with little forethought concerning the programming languages or software packages which will be used for processing or analysis. For this reason, unusual values such as 99's may be coded for missing values. The example which follows demonstrates using RECODE to replace 99's and -1's by numeric missing values for the variables Q1 and Q20 through Q30. The keyword _ALLNUM_ is used to replace the character values '9' and '-' with blanks for all alphanumeric variables. All other values in the data set remain unchanged.

```
PROC RECODE OUT=EDITED;
RECODE Q1 Q20-Q30 99=-1.,
_ALLNUM_ (EXCEPT AGE WEIGHT Q1)
ELSE=99;
```

Example 2: Data are often keyed as numeric for convenience and to save punching time. RECODE provides an easy mechanism for replacing numeric values with character values when labeled output is desired. The example which follows also demonstrates the ease with which a range of values may be collapsed into a single character value.

```
PROC RECODE DATA=Numeric OUT=Labeled;
RECODE RACE='WHITE' 2='BLACK' 3='OTHER'/
ELSE='';
RECODE PROFCODE 100 THRU 199='NURSES AIDES'
200 THRU 299='LICENSED PRACTICAL NURSES'
300 THRU 399='REGISTERED NURSES'
LO THRU 50='CLERICAL STAFF';
RECODE _ALLALPHA_ (EXCEPT SSN) 09=1 10=2
11=3 'W'=1 'B'=2 'O'=3 'E'=5='STRONGLY AGREE' 6='AGREE'
7='NO OPINION' 8='DISAGREE'
9='STRONGLY DISAGREE' / ELSE='MISSING';
```

Example 3: RECODE could be particularly helpful in the processing of optical reader output files. Such files often contain special characters to signal invalid data or multiple responses. The example which follows demonstrates how easily these values can be replaced by SAS missing values. The legitimate responses are easily changed to numerics with the CONVERT option.

```
PROC RECODE;
RECODE ALL (EXCEPT SSN) 'O'=., 'I'=.
/ CONVERT;
```

Example 4: RECODE's ability to edit character strings is useful when prefixes occur in values. This feature can be used to collapse categories or to make substitutions at the front of existing values.

```
PROC RECODE;
RECODE BODYTYPE '12A':='FORD MUSTANG':
'16B':='VW RABBIT':
'23C':='TOYOTA CORONA':;
RECODE PART '42-':='WHEEL' '36-':='ALTERNATOR';
```

RECODE has a logical and simple syntax, but it is a powerful procedure with potential usefulness in a large number of different situations. The availability of RECODE will aid substantially in report generation, data editing, and day-to-day data manipulation. It is evident that this procedure is useful, but its full range of capabilities will be realized only after users have applied it to their own particular needs.