THE DATA BASE MANAGEMENT SYSTEM

USER MANUAL AND EXAMPLE

by

Michel J. Bastarache
Ernest Allen Hershey III

ISDOS Working Paper No. 89

Preliminary Draft

April 1975

ISDOS Research Project
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48104
(313/763-3469)

This is a working paper and should not be quoted or reproduced in whole or in part without the written consent of the authors. Comments are solicited and should be addressed to the authors.

ensn

UMR0276

Preface

The purpose of this paper is to serve as an aid in using the DBMS subroutines specified by Ernest Allen Hershey in ISDOS Working Paper No. 88, "A Data Base Management System for PSA Based on DBTG 71." It is assumed that the reader is familiar with the contents of this paper and that it is available for reference. A thorough understanding of FORTRAN is also assumed. This Working Paper specifies the use of Version D2.0 of the Data Base Management System (FORTRAN Implementation). Note that data bases generated by Version D1.0 (Assembler Implementation) are not compatible with Version D2.0.

Table of Contents

Preface i

Introduction 1

1. Overview of the System 4

1.1 Background 4

1.2 Implementation 4

1.3 Hardware and Software Considerations 5

1.4 Logical and Physical Data Structures 5

1.5 System Functions 5

2. Specifying a Data Structure 8

2.1 Grouping Data into Records 8

2.2 Relating Records 9

2.3 Contents of the Presidential File (PRESFILE) 11

2.4 A Data Structure for the Presidential Data Base 16

3. Specifying the DDL for a Data Structure 20

3.1 Naming Conventions 20

3.2 Records and Items 20

3.3 Item Size 25

3.4 Handling Relationships Among Records 25

3.5 Handling Groups 27

3.6 Handling Repeating Items 28

3.7 Handling Repeating Groups. 28

3.8 Specifying Data Base Size 29

3.9 Developing the DDL for the Presidential Example 29

4. Creating and Initializing the Data Base 37

4.1 Creation of DDL, DBTF and DBF Files 37

Table of Contents (cont'd)

4.2 Generation of the DBTF 37

4.3 Initialization of the DBF 37

5. Coding Conventions for Data Base Access Programs 38

5.1 Common Areas and Block Data 38

5.2 Buffer Variables 41

5.3 Access of the DBF and DBTF 41

5.4 General Program Formats 42

6. Populating the Data Base 43

6.1 Reading Data from the Input File 43

6.2 Storing Data 44

6.3 FMSK 44

6.4 CR 44

6.5 SFR 45

6.6 AMS 45

6.7 SRM 45

6.8 SOM 46

6.9 SMOVE 46

6.10 Code to Populate the President Data Base 47

7. Generating Reports from the Data Base 70

7.1 Retrieving Data 70

7.2 FFM 70

7.3 FNM 70

Table of Contents (cont'd)

7.4 SOM 71

7.5 SMM 71

7.6 GFM 71

7.7 GFO 72

7.8 Report Routines 72

7.9 HEDING. 72

7.10 BUFBLD 72

7.11 NINBUF 73

7.12 PBUF 73

7.13 Code to Generate Reports from the Presidential Data Base . . 73

Appendix A Presidential File (PRESFILE) Listing 85

Appendix B Presidential Report Examples 96

INTRODUCTION

Before any attempt is made to explain the implementation of this Data Base Management System (DBMS), definitions for the terms to be used must be presented. The following terms will be used throughout the paper.

type (vs. occurrence)
item
group
record
set
DDL
DML

type (vs. occurrence) When describing various objects in this paper it is necessary to differentiate between a "type" of object (i.e., its class) and an instance of one object within a "type" (i.e., an "occurrence"). For example, Social Security number may be a "type" of item, but an "occurrence" of a Social Security number might be 366-60-5321. This is a very important concept and comprehension of the difference is essential.

item The "item" is the elementary data unit from which all other types of structures are ultimately composed. An occurrence of an "item" is the representation of a value which may be a number, a string of characters, a truth value, etc. Examples of types of items are age, hair, color, occupation, etc. Occurrences of these types of items might be: 35, BROWN and TEACHER, respectively. (An item is often called a field, data item or element in other DBMS's.)

group A "group" is a named collection of "items" and/or other "groups." An occurrence of a group is an instance of a value for each item contained within it. Examples of types of possible groups are name (which might consist of first name, initial and surname), and date (consisting of month, day, year). Occurrences of these types of groups might be: JOHN Q. ADAMS and January 1, 1972, respectively. (A group is often called a segment or data aggregate.)

record A "record" is a named collection of "items" and/or "groups." There may be an arbitrary number of occurrences in the data bases of each record type. The record is used to represent the major entities of an application. For example, if the system being constructed were a payroll system, it is very likely that there would be an occurrence of an EMPLOYEE-RECORD for each employee on the payroll.

set A "set" is a named collection of "records" which specifies an ordering or relation among the record types within it. The specification of "owner records" and "member records" within the set designates the direction of the relationship. For example, if records were assigned to correspond to each person and each state in the U.S., a relationship between the two record types might be defined to associate all those people born in the same state. Since a person can only be born in one state and states can have several people born in them, the direction of the relationship is specified by designating state records as owner records and people records as member records (i.e., for any occurrence of a state record, there can be many occurrences of member records in the relationship.)

DDL The "DDL" (Data Definition Language) is a language used to define data structures within a data base, usually based on terminology similar to that specified previously.

DML The "DML" (Data Manipulation Language) is a language used to manipulate the data in the data base defined by the DDL. (For this particular system they are the subroutines described in Working Paper No. 88.)

The example that will be referenced throughout this paper is called the "Presidential Data Base." The input to our problem is the Presidential File (PRESFILE) which contains data about each President, from George Washington to Lyndon Johnson. Our goal is to implement a system which will store all this data into a data base in a form easy for retrieval. Finally, reports will be generated from the contents of the data base. The following types of data are contained in the PRESFILE and must be incorporated into the data base:

President data

Administration data

Congress data

Election data

State data

If one type of record could be defined to correspond to each of these data types, the data structure might look like Figure 1:

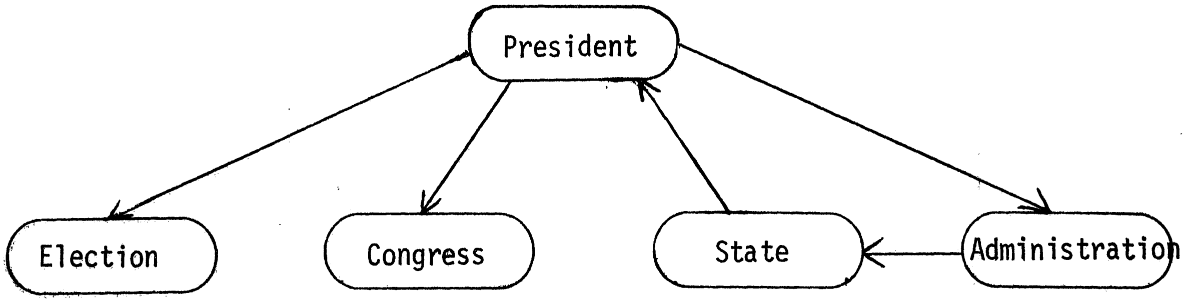


Figure 1

Any given President (occurrence of a President record) might be related to several Elections (occurrences of Election records) by the fact that the President won those elections. (By the previously defined terminology, the President records can be defined to be "owner records" for a "set" where the "member records" are Election records. This "set" could be called ELECTION-WON.) By the same token, a State record could be related to one or more President records by the fact that they were born there, and one or more State records could be related to Administration records by the fact that those states were admitted to the Union during that Administration, etc.

These relationships can be specifed more clearly by a DDL format;

```
SET Elections-Won
   OWNER President
   MEMBER Election

SET Presidents-Born
   OWNER State
   MEMBER President

SET Administrations-Held
   OWNER President
   MEMBER Administration

SET States-Admitted
   OWNER Administration
   MEMBER State
```

Notice that all the relationships (sets) have been defined so that for any given occurrence of an owner record, there exists one or more occurrences of a member record (which can be notated as a 1:M relation). Utilizing this convention makes the task of specifying the DDL that much easier.

Each record type defined (President, Election, etc.) will consist of smaller collections or units of data. The Presidential record may contain information about date of birth, age, etc. Date would be defined as a "group" as it could be broken down into the "items": month, day and year. Depending on the facilities allowed in the DDL. specification of groups may or may not be allowed. The DDL presented in this system does not allow for groups, so all data within the Presidential record must be defined in terms of "items."

1. OVERVIEW OF THE SYSTEM

1.1 Background

This system was designed based on the specifications given by the CODASYL Data Base Task Group in their "CODASYL-COBOL Data Base Facility Proposal". It was developed and implemented by Ernest A. Hershey at the University of Michigan in the summer of 1973. Since then the system has been used in several independent applications:

Problem Statement Analyzer - The DBMS is used to maintain the data bases needed for the PSL/PSA System. (See ISDOS Working Paper No. 86 for more information about this application.) *

Computerized Bibliography - The DBMS is currently being used to maintain a data base consisting of an up-to-date bibliography for the ISDOS Project. *

Tape Library System - The DBMS is being used to maintain a tape library for another project at the University. This particular system has been set up on Michigan's 370/168 as well as on a Honeywell machine.

Presidential Data Base Example - To illustrate some of the capabilities of the DBMS this example has been implemented based on the data presented by A. Vorhaus and A. Weinert of the CODASYL Systems Committee. *

1.2 Implementation

This system is a host langauge Data Base Management System (DBMS). A system with host language capability is one which is built upon the facilities of a procedural language such as FORTRAN. The DBMS functions** are invoked from within the host language program written in COBOL or FORTRAN. The method of interface between system and program (be it either FORTRAN or COBOL) is via the CALL statements in the program. For this reason, the user must be somewhat experienced in programming. The user of the system is considered to be a programmer in the sense that a set of FORTRAN (or COBOL) statements must be written to be executed sequentially. The user exercises the same degree of procedural control over processes as if programming in the host language except that the facilities of the DBMS handles data transfers to and from the data base. This also means that all the capabilities of the host language are at the user's disposal. The high level DBMS functions such as UPDATE, DELETE, etc., must also be written by the user for a particular implementation of the system.

* This application is currently operative on the IBM 370/168 at the University of Michigan.

** In this sense "functions" means retrieval, storage, etc.

1.3 Hardware and Software Considerations

The DBMS described in this paper can be set up on any machine which has an ANSI FORTRAN or ANSI COBOL compiler. This makes the DBMS machine independent and operating-system independent. Mode of operation (on-line or batch) is of course dependent on the facilities of the installation.

1.4 Logical and Physical Data Structures

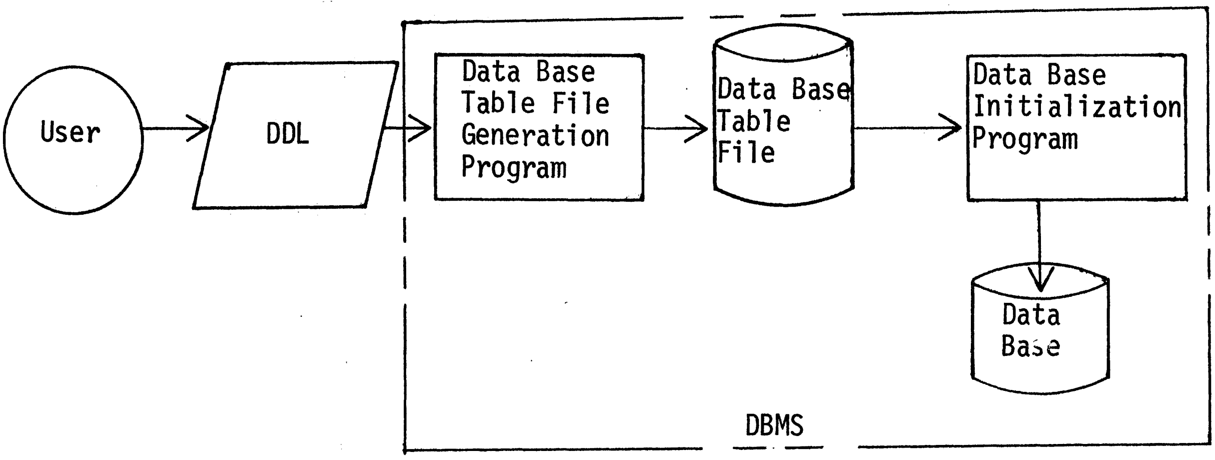
All types of logical data structures, from simple lists to complex network structures may be defined when using the DBMS. The Presidential data base example presents an implementation of a network structure.

Any logical data structure defined by the user is translated into a doubly linked list structure for each "set" defined in the data base. Records are stored according to a "best fit" algorithm and are identifiable by a unique "data base key". The number of records and sets that can be handled by the DBMS is, for all practical purposes, limited only by the amount of storage available.

1.5 System Functions

The method of defining the data base is shown in Figure 1c.

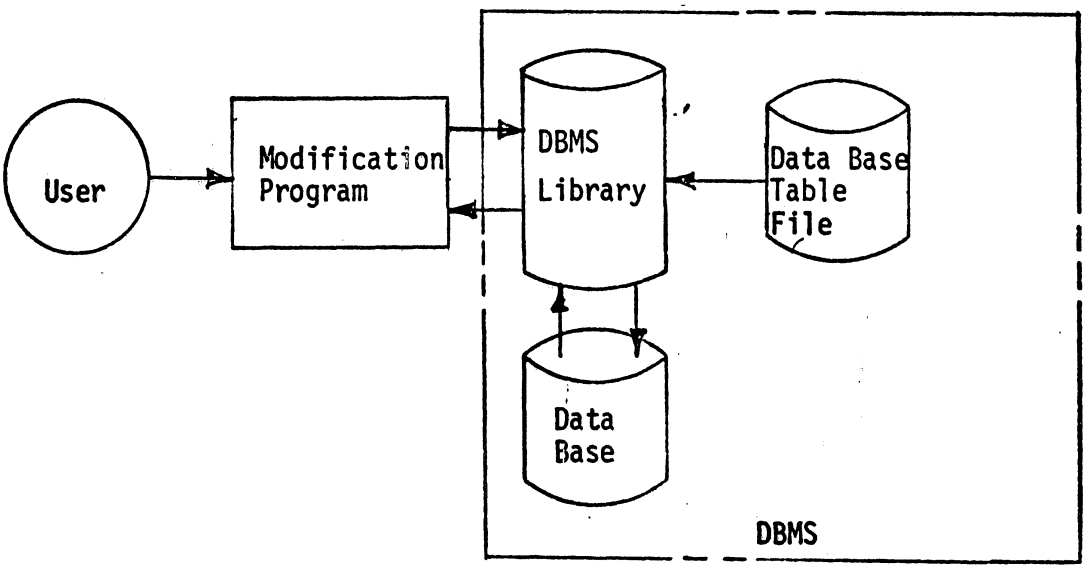
1a. Interaction of the DBMS with the user in initializing the data base:


User
DDL
Data Base Table File Generation Program
Data Base Table File
Data Base Initialization Program
Data Base
DBMS

The user must specify the DDL for a particular usage of the system. This DDL is then to be used as input to a program which formulates a set of tables. These tables are used as input to another program which prepares the data base for accepting data. The tables are also used when referencing the data base for modification and retrieval purposes.

The method of populating the data base or modifying it in any way is pictured in Figure 1b.

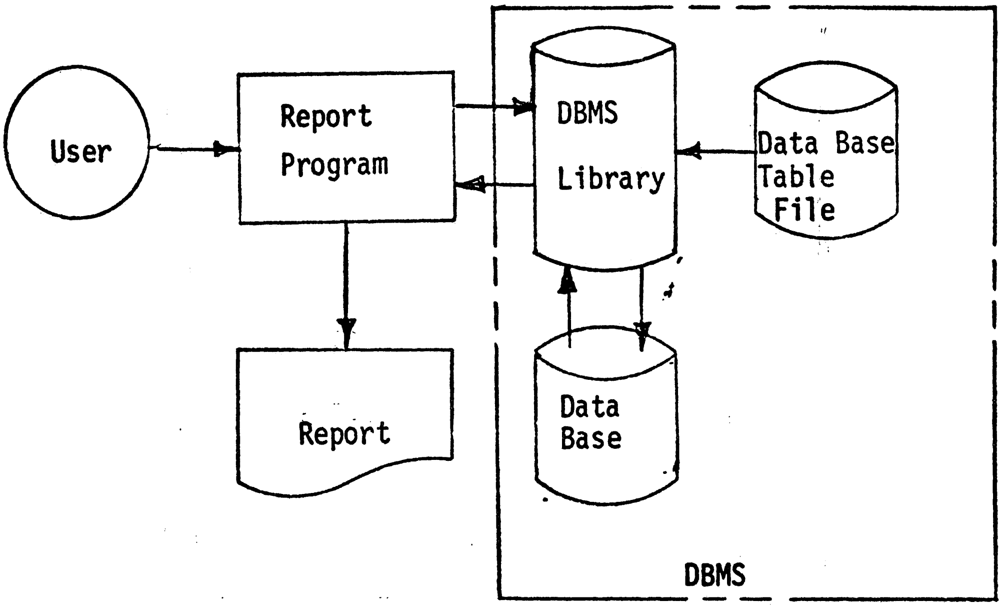
1b. Interaction of the DBMS with the user in modifying the contents of the data base:



The user writes a program utilizing routines in the DBMS library. These routines also provide the means of interfacing the user's program with the Data Base Table File (DBTF), generated in the creation phase, and the data base. Error messages are generated by the DBMS routines should errors occur in modifying the data base.

The method of retrieving data from the data base in the form of reports is shown in Figure 1c.

1c. Interactionof the DBMS with the user in generating reports from the data base:



The user must write a program which calls routines in the DBMS library. These routines provide the means of referencing the DBTF for access to the data base and retrieving the data. The errors encountered while retrieving data will be documented on the user's outputs.

2. SPECIFYING A DATA STRUCTURE

Before the user writes the DDL for a particular system some preliminary considerations must be taken into account. These are: how is the data to be conceptually grouped(what will be defined to be records) and how will these groupings relate to each other. The users conception of how this data is related (as opposed to how it is actually defined in the DDL or how it is physically stored) is called the "data structure".

2.1 Grouping Data into Records

The first decision to be made is how to group the data. This can be determined after studying the requirements for the system. In most cases grouping of data is trivial as all the data pertaining to one type of object (such as a person) would be grouped together. For example, if your system was concerned with maintaining information about people, it would be very likely that data used in identifying each person would be grouped together (i.e., name, social security number, etc.)

When it is not obvious whether to add data to one defined record or define a new one to contain it, the following guidelines may be helpful.

i. Attempt to reduce the degree of redundancy in the data structure (i.e., whenever possible store data in only one record rather than two).

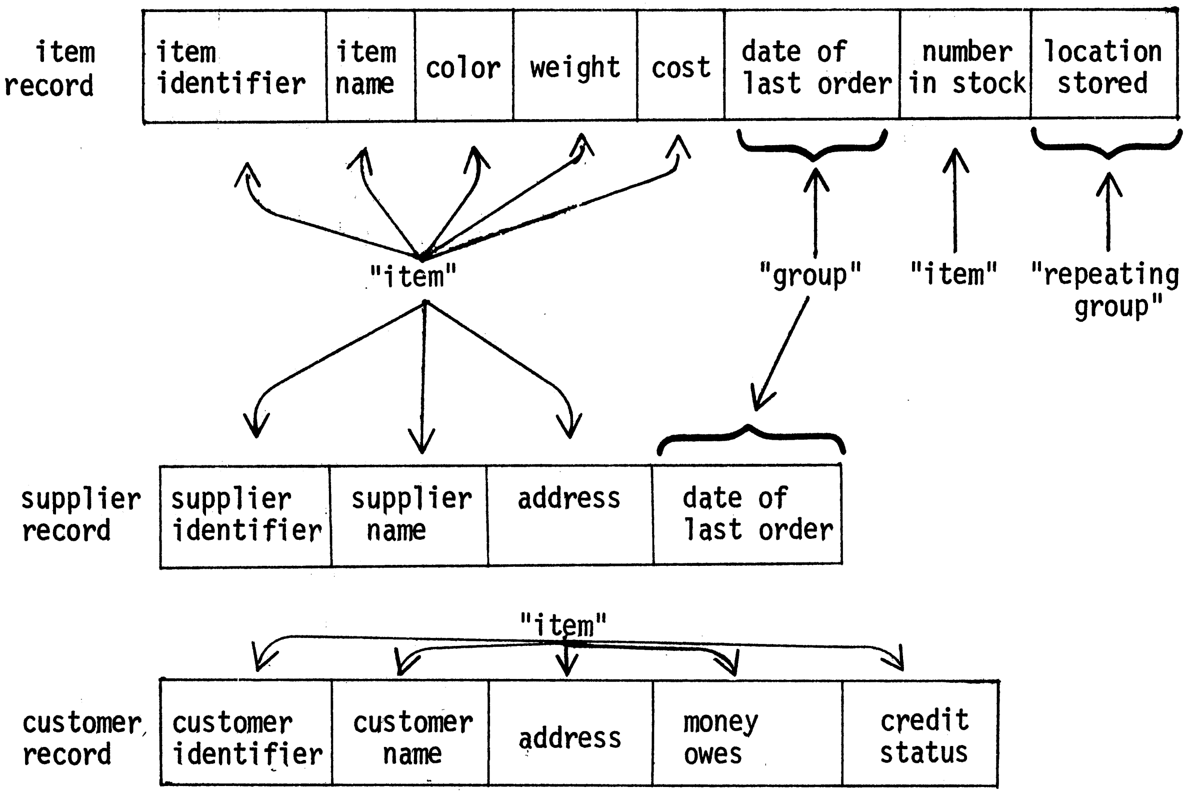
ii. Define records (and the data within them) according to the requirements for retrieving this data. For example, if an inventory system were being developed and one of the output requirements was to generate a report consisting of all suppliers of items in the inventory, it would be a good idea to have a record type associated with suppliers (as opposed to storing the supplier information as part of the records for each item in the inventory.

After determining what data is to be contained in which records, a decision must be made as to how this data is grouped within the record, if at all. For example, it may be convenient to think of all data pertaining to a person's family as MARRIAGE-DATA which could be defined then as a "group" of items (i.e., marriage date, spouse's name, number of children, etc.).

If several occurrences of a particular group were possible for a single record occurrence (as in the case of MARRIAGE-DATA where a person could be married more than once) this type of data would be called a "repeating group". Likewise, a "repeating item" is a simple case of the repeating group where the group consists of only one item (e.g., OCCUPATION).

The end result of the data grouping process is a number of defined record types and specification of the data within them in terms of items, groups, repeating items and repeating groups for each record type.

For a simple inventory system the result could be shown by the following format.



item record	item identifier	item name	color	weight	cost	date of last order	number in stock	location stored

supplier record	supplier identifier	supplier name	address	date of last order

customer record	customer identifier	customer name	address	money owes	credit status

Note that most data could be defined as "items". "Date of last order" was defined as a "group" since it consists of month, day and year and "location" is a "repeating group" because any given inventory item might be stored at more than one location and a given location might be identified by warehouse number, row number, rack number, etc.

2.2 Relating Records

The next step in specifying the data structure is to define the relationships among the records defined.

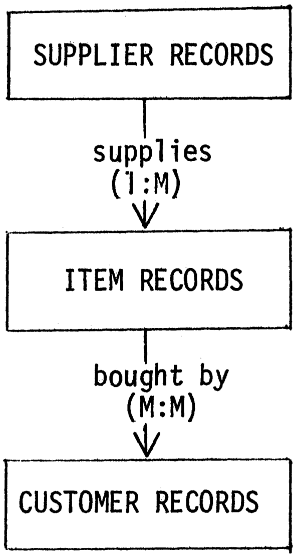
Between any two records that are to be related the type of relationship should be determined as 1:1, 1:M or M:M.

1:1 means that for every occurrence of a particular record type there is one and only one occurrence of another record type related to it. For example, all currently married men and women in the United States could be paired by the relationship of "marriage". Since any of these people are allowed only one spouse, the relationship would be designated as a 1:1 relationship.

1:M means that for every occurrence of a particular record type there can be zero, one or many occurrences of another record type related to it. In the inventory example, this type of relationship exists between the supplier and item records. A particular supplier could supply one or more of the items in the inventory.

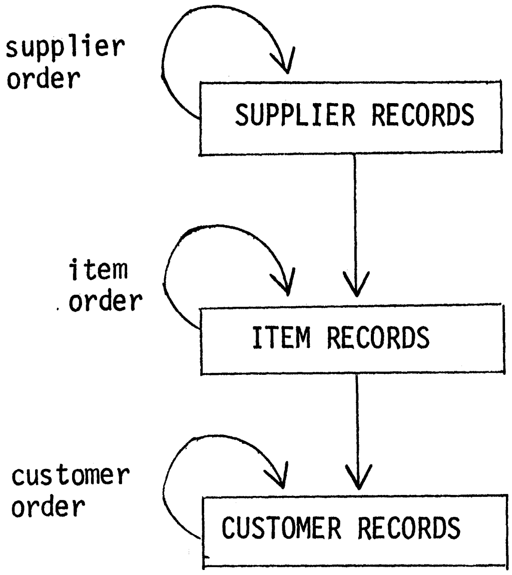
M:M means that for an occurrence of a particular record type there can be many occurrences of a second record type related to it and likewise any occurrence of the second record type could be related to several occurrences of the first record type. This type of relationship is apparent when relating the item records to the customer records. Several items can be bought by one customer and several customers could buy the same type of item.

The relationships for the inventory example can be shown in the following format.



The titles in lower case present a name for the relationship and the arrow specifies the direction. The "type" of relation is contained in parentheses. This diagram then says that a particular supplier "supplies" one or more items in the inventory which are "bought by" many customers.

A relation which specifies an order among the record occurrences of one or more record types is called an ordering relationship. Specifying this type of relationship aids in retrieving particular record occurrences and in formatting any output reports (the data is already ordered). Ordering relationships for the inventory data structure can be shown in the following way.



In most cases a particular "item" in the record will be used to designate order (the identifier "item" usually, but order could just as well be established on weight or cost for inventory items records).

2.3 Contents of the Presidential File (PRESFILE)

Before attempting to specify a data structure for the Presidential Data Base an inspection of PRESFILE must be made to see what data is available. The complete listing of all the data in PRESFILE is given in Appendix A. In summary, PRESFILE presents five different types of data groupings which are identified by the names:

PRES

ELECTION

ADMIN

CONGRESS

STATES

Following each identifier is a collection of data in a specific format, representing one occurrence of a data type. PRES, for example, designates that all following data (up to the next identifier) presents information about a particular president.

ELECTION designates that the following data pertains to a particular election; ADMIN, to a particular administration; CONGRESS to a particular congress; and STATES to a particular state.

The format of the data * following a PRES identifier is given below with respect to an actual example:

PRES	data identifier	
WASHINGT	President-identifier (usually an abbreviation of the last name)	
WASHINGTON	last-name	President's name
GEORGE	first-name	
	middle-initial	
FEBRUARY	month	birthdate
22	day	
1732	year	
VIRGINIA	state-born-in	
VIRGINIA	state-identifier	(identifier name for the state born in)
6 FT. 2 IN.	height	
FEDERALIST	party	
	college	
ENGLISH	ancestry	
EPISCOPAL	religion	
3	#-of-occupations	
SURVEYOR	occupations	(number of occupations is determined by value of #-of-occupations)
FARMER		
SOLDIER		
DECEMBER	month	death-date
14	day	
1799	year	
PNEUMONIA	cause-of-death	
AUGUSTINE	father-name	
MARY	mother-name	
1	#-of-marriages	

* All the data in PRESFILE, including identifiers, were restricted to ten character representation thus resulting in abbreviations of some names.

MARTHA	wife		marriage-data	(number of occurrences of marriage-data is determined by value of #-of-marriages
JANUARY	month	marriage-date		
6	day			
1759	year			
0	number-of-children			
2	#-of-elections-won			
E1789		election-identifiers		(number of elections is determined by value of #-of-elections-won)
E1792				
2	#-of-administrations-headed			
A1		administration-identifiers		(number of administrations is determined by value of #-of-administrations-headed.)
A2				
4	#-of-cabinets			
C1		cabinet-identifiers		(number of cabinets is determined by value of #-of-cabinets.)
C2				
C3				
C4				

Note that any fields which are not defined (such as middle initial and college in this example) are represented by ten blanks.

The format of the data following an ELECTION identifier is given below:

ELECTION	data identifier		
E1789	election-identifier		
1789	election-year		
WASHINGTON	winner		
FEDERALIST	winning-party		
69	winning-votes		
1	#-of-opponents		
ADAMS	opponent	opponent-data	(number of occurrences of opponent-data is determined by value of #-of-opponents.)
FEDERALIST	opponent-party		
34	opponent-votes		

The format of the data following an ADMIN identifier is given below:

ADMIN	data-identifier		
A1	administration-identifier		
APRIL	month	inauguration-date	
30	day		
1789	year		
WASHINGT	president-identifier		
1	#-of-vice-presidents		
JOHN	first-name	vice-president-name	(number of occurrences of vice-president name is determined by value of #-of-vice-presidents.)
ADAMS	surname		
2	#-of-states admitted		
VERMONT	state-name	state-data	(Number of occurrences of state-data is determined by value of #-of-states-admitted.)
VERMONT	state-identifier		
KENTUCKY	state-name	state-data	
KENTUCKY	state-identifier		

The format of the data following a CONGRESS identifier is given below:

CONGRESS	data-identifier
C1	Congress-identifier
2	#-of-Senate-parties

FEDERALIST	Senate-party	Senate-data	(number of occurrences of Senate-data is determined by value of #-of-Senate-parties)
17	number-of-Senators		
ANTI-FED	Senate-party	Senate-data	
9	number-of-Senators		
2	#-of-House-parties		
FEDERALIST	House-party	House-data	(Number of occurrences of House-data is determined by the value of #-of-House parties.)
38	number-of-members		
ANTI-FED	House-party	House-data	
26	number-of-members		

The format of the data following a STATES identifier is given below.

STATES	data-identifier		
ALABAMA	state-identifier		
ALABAMA	state-name		
1814	year-admitted		
MONTGOMERY	capital		
51609	area		
29	area rank (as compared to all 49 other states)		
3556000	population		
21	population-rank (as compared to all 49 other states)		
10	number-of-House-votes		
4	#-of-major-cities		
BIRMINGHAM	city-name	city-data	
340887	city-population		
MOBILE	city-name	city-data	(The number of occurrences of city-data is determined by the value of #-of-major-cities.)
202779	city-population		
MONTGOMERY	city-name	city-data	
134393	city-population		
HUNTSVILLE	city-name	city-data	
123519	city-population		

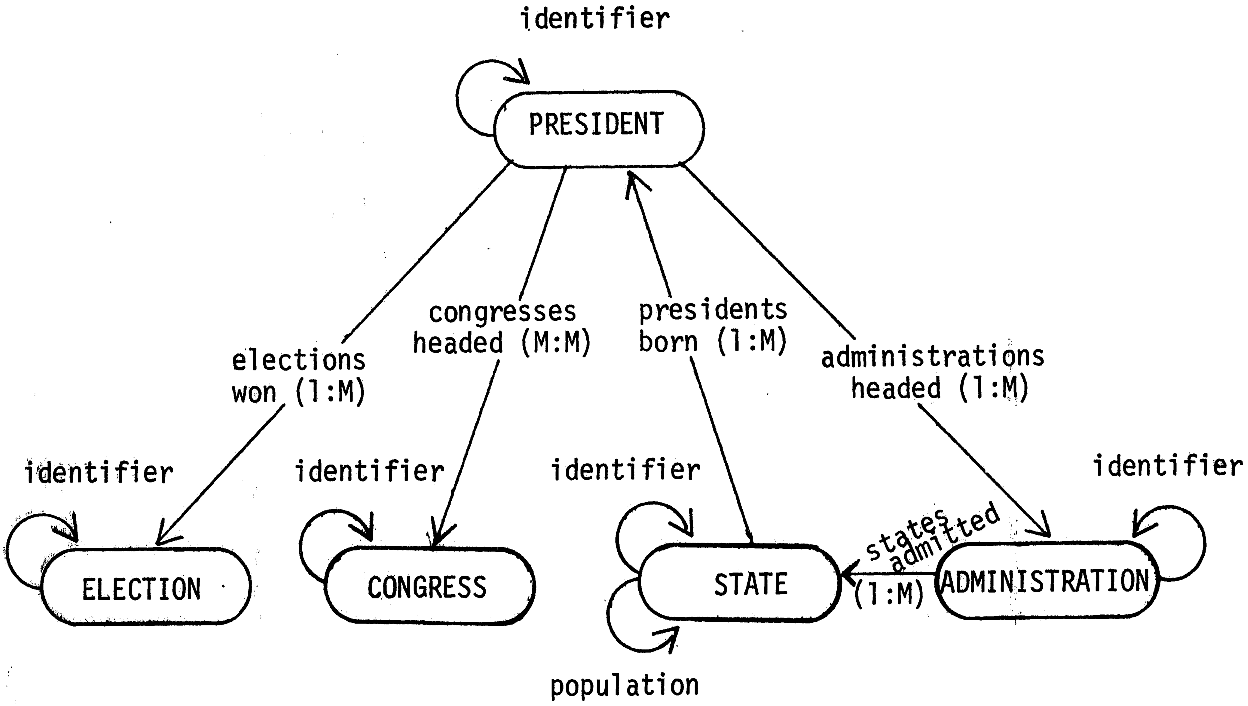
Notice that state-identifier state-name are usually the same. The major difference between the two is that blanks and special characters are allowed in a state-name but not for state-identifier. For example, "NEW YORK" is the name of the state, but "NEWYORK" is the identifier for the state. (This was a restriction imposed by the PRESFILE format rather than the DBMS.) The complete listing of PRESFILE can be found in Appendix A.

2.4 A Data Structure for the Presidential Data Base

Now that all the data available to the problem is known, an attempt can be made to structure it.

First off, a record type can be defined to correspond with each of the five possible data group types (PRES, ELECTION, ADMIN, CONGRESS and STATES). Some of the data in these groups are redundant (occur in more than one group) and so an attempt to eliminate redundancy will be made by specifying relationships between records. For example, within the group of data describing a President, the state he was born in and the identifier for that state are given. This data could be eliminated altogether if a relation was set up between the President record and State record specifying that the particular President was born in that State.

After eliminating redundant data the following relationships are given between record types.



Note that all record types are ordered by identifier for retrieval purposes. STATE is also ordered by population size.

The remaining task to be performed is to identify the data within each record type and identify each piece of data as an "item," "group," "repeating item" or "repeating group." Though the contents of the record types are very similar to the contents of the data groups specified in the PRESFILE it will be presented here for the sake of clarity.

President Record --

Data represented			Data type
President-identifier			item
First-name (item)		President-name	group
Middle-initial (item)			
Surname (item)			
month (item)		Birthdate	group
day (item)			
year (item)			
Height			item
Party			item
College			item
Ancestry			item
Religion			item
month (item)		Death date	group
day (item)			
year (item)			
Father			item
Mother			item
#-of-occupations			item
occupation			repeating item
Wife (item)		Marriage-data	repeating group
month (item)	Marriage date*		
day (item)			
year (item)			
number-of-children (item)			

* Marriage-date is a group within a repeating group.

Election Record --

Data represented		Data type
Election-identifier		item
Election-year		item
Winner		item
Winning-party		item
Winning-votes		item
Opponent (item)	Opponent-data	repeating group
Opponent-party (item)		
Opponent-votes (item)		

Administration Record --

Data represented		Data type
Administration-identifier		item
month (item)	inauguration-date	group
day (item)		
year (item)		
first-name (item)	vice-president-name	repeating group
surname (item)		

Congress Record --

Data represented		Data type
Congress-identifier		item
party (item)	Senate-data	repeating group
members (item)		
party (item)	House-data	repeating group
members (item)		

State Record --

Data represented	Data type
State-identifier	item
State-name	item
Year-admitted	item
Capital	item
Area	item

State Record (Continued)

Data represented		Data type
Area-rank		item
Population		item
Population-rank		item
Electoral-votes		item
City-name (item)	City-data	repeating group
City-population (item)		

3. SPECIFYING THE DDL FOR A DATA STRUCTURE

Now that a data structure has been specified, this structure must be defined in DDL statements for processing by the DBMS. The conventions of translating the data structure into its equivalent DDL are given in the rest of this section. The only objects that can be defined by the DDL are RECORDS, ITEMS, REPEATING ITEMS, and SETS. All data described in the data structure must consequently be defined in these terms. Refer to Figure 1, "CARD FORMATS," and Figure 2, "CARD ORDERING RESTRICTIONS," for proper syntax of the DDL.

3.1 All data names defined by the user are restricted to a six character representation. For example, in the Presidential Data Base Example "President-identifier" and "college" were two names representing items types in the data structure. Because of the six character restriction these names must be truncated or abbreviated. "President-identifier" could be called IDENT and "college" COLEGE. Names of record types, however, must be unique for the entire system. This means that two item types could have the same name "IDENT" as long as they were contained in two different record types. Two record types of the name "PRES" would be illegal.

3.2 Records and Items

Since records and items are objects allowed in the DDL terminology, the records and items defined in the data structure may be translated directly into DDL statements. For example, the data for the Election record could be defined as such:

```
RECORD  ELECT

ITEM    IDENT    CHAR  10

ITEM    YEAR     INTEG 31

ITEM    WINNER   CHAR  10

ITEM    PARTY    CHAR  10

ITEM    VOTES    INTEG 15
```

The repeating group in the Election record must be handled in a special way and will be described later. Notice that each item is defined to be either "CHAR" or "INTEG." This specifies what type of data is contained within the item; a character string or an integer number, respectively. There are six possible types of allowable data:

INTEG	(integer number)
REAL	(real number)
BINARY	(binary number)
DBKEY	(data base key value)
LOGIC	(logical value)
CHAR	(character string)

CARD FORMATS

RECORD card:

col 1 - 6	RECORD
col 8 - 13	record type name

ITEM card:

col 1 - 4	ITEM
col 8 - 13	item name
col 15 - 20	item type legal types: INTEG, REAL, BINARY, DBKEY, LOGIC, CHAR
col 22 - 24	item size legal size:

type	maxsize	units
INTEG	31	bits
REAL	63	bits
BINARY	256	bytes
DBKEY	-	always 4 bytes
LOGIC	-	always 4 bytes
CHAR	256	bytes

col 26 - 31	depended on item (optional)
col 33 - 35	max replication factor

SET card:

col 1 - 3	SET
col 8 - 13	set type name
col 15 - 20	order legal orders: FIFO, LIFO, NEXT, PRIOR, IMMAT, SORTED
col 26 - 31	sort key (if order = SORTED)

OWNER card:

col 1 - 5	OWNER
col 8 - 13	owner record type name

MEMBER card:

col 1 - 6	MEMBER
col 8 - 13	member record type

NPAGES card:

col 1 - 6	NPAGES
col 22 - 24	data base size (in pages)

Figure 1

CARD ORDERING RESTRICTIONS

I. Record Descriptions

1. Items can only be defined in relation to the record they belong to (i.e. an item cannot be independent).

2. The items which make up a record type must be described by ITEM Cards and immediately follow the appropriate RECORD Card.

3. The order in which ITEM cards occur after the RECORD Card determines the order in which the data is physically stored in the record.

4. A repeating item is defined when the "max replication factor" field is assigned a value.

5. If a replication factor is given for an item, it is assumed to repeat.

6. A repeating item may only be the last item in a record.

7. The "depending on item" field for a repeating item designates an item (defined elsewhere in the record description) whose value specifies the number of occurrences of the repeating item (for a particular record occurrence).

8. A depending on item must be of type INTEG.

9. An item must be defined previously in the record before it can be used as a depending item in the record.

10. The depending on item is only checked for if a "max replication factor" is given.

II. Set Descriptions

1. There must be at least one owner card and one member card for each set.

2. If SYSTEM is given as an owner record type name, no other owner cards may be given for that set.

3. The owner card(s) for a set must come immediately after the SET card for the set.

4. The member card(s) for a set must come immediately after the OWNER card(s).

5. A record type must have been defined previously (by a RECORD card) before it can be used as an owner or member type.

Figure 2

II. Set Descriptions (Continued)

6. If a sort key is given for a set, it must be an item previously defined (in a record description) and be of the same type, size and displacement in all member record types.

7. A sort key item may only be of type CHAR or INTEG.

Figure 2 (Continued)

IBM/360-370 STORAGE CHARACTERISTICS

Item type	Size defined in DDL	Allotted Storage Space	Range of values represented
INTEG	1-31 bits	full word	-2147483648 thru +2147483647
REAL	1-31 bits	full word	$\pm$ (16^{-65} thru 16^{63}) **
REAL	32-63 bits	double word	$\pm$ (16^{-65} thru 16^{63}) **
BINARY	$1 \leq n \leq 256$ bytes	N words *	$-2^{(32\cdot N)-1}-1$ thru $+2^{(32\cdot N)-1}-1$
DBKEY	(always 4 bytes)	full word	(not relevant)
LOGIC	(always 4 bytes)	full word	True or False
CHAR	$1 \leq n \leq 256$ bytes	N words *	0 thru 4·N characters

* Where $N = f(\frac{n+3}{4})$ [f rounds the expression to the lowest integer]
For example, if n=10: $N=f(\frac{10+3}{4})=3$ full words.

** This range is approximately $.5397605 * 10^{-78}$ through $.7237005 \times 10^{76}$ for both single (full word) and double precision (double word). Double precision, however, allows for seven additional digits of accuracy over the single precision equivalent. Of course the representation of the zero value 0.0 can be represented as a real number in addition to these value ranges.

Table 1

The maximum size of the data value in each item is specified as either "10", ten bytes (characters) or "15" and "31" bits (integer). Table 1 specifies the ranges of allowable size for each of the six above data types.

3.3 Item Size

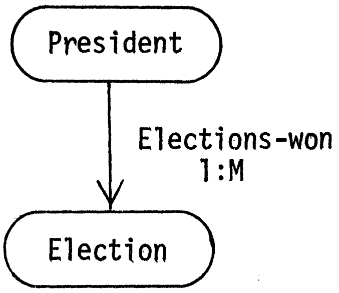
The DDL allows the user to specify a size for each item defined. Regardless of the size given, however, the data must be stored as a full word or some integer multiple of full words. Any unused space in the full word will be packed with an appropriate filler (e.g., blanks if the value is a character string and zeros if the value is an integer.) If a size of 6 bits were given for an item defined to be of type INTEG, each occurrance of this item would be reserved a full word of storage. The size here is only of relevance to the user as it designates some restriction to the range of values which should be accepted by the user's program. The DBMS makes no distinction if the size is 15 or 31 bits (in the case of integers) since an entire fullword is saved to store any integer value.

Treatment of item size for character strings (type CHAR) or bit strings (BINARY) is a bit different than that of type INTEG. Table 1 presents a method of calculating the number of full words of storage assigned to each occurrence of an item (as defined by its type and size). It also presents the range of values allowed for a particular item (as defined by its type and size). Note that the range of values is directly dependent on the allotted storage space.

3.4 Handling Relationships Among Records

It is a simple matter to define record relationships in the data structure as SETs in the DDL.

For relationships defined as 1:1 or 1:M in the data structure (take the relationship between President records and Administration records for example):



The President record will be denoted as PRES in DDL terminology, Election record as ELECT and the Elections-won relationship as PRESEN. The following algorithm can be used:

i. Choose a name for the relationship (it must be unique). (In this example the name used is PRESEN.)

ii. The record type at the tail of the relationship is designated as the OWNER record.

iii. The record type (or types) at the head of the relationship is designated as a MEMBER record.

iv. All the occurrences of member record types to a particular occurrence of an owner record can be ordered in some way if desired (See below.)

The DDL needed to represent the Elections-won relationships is given as:

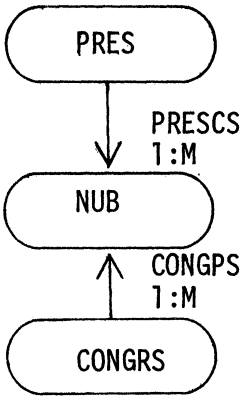
```
SET          PRESEN       SORTED       IDENT
OWNER        PRES
MEMBER       ELECT
```

The "SORTED IDENT" phrase means that all member recora occurrences for a particular owner record occurrence will be sorted on the item, IDENT, within the member record. This ordering is for search and retrieval purposes.

For a relationship defined as M:M the task of specifying this in the DDL is a little more complex. (The relationship between Congress and President records will be used as the example.)

The President record is denoted as PRES in the DDL, the Congress record as CONGRS and the two relationships that will be defined are PRESCS and CONGPS. Also, another record type must be defined which will be called NUB.

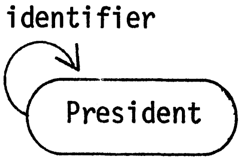
The data structure will be changed to the following by the use of the NUB record:



The DDL for this is specified in the same way as for two 1:M relationships:

```
SET          PRESCS       FIFO
OWNER        PRES
MEMBER       NUB
SET          CONGRS       IMMAT
OWNER        CONGRS
MEMBER       NUB
```

To specify an ordering relationship in the DDL, one item in the record type to be ordered must be chosen as the sort key. Take the example:



from the data structure. This specifies that the President record (PRES) is to be ordered by its identifier item (call it IDENT). This can be specified by the DDL in the following manner:

```
SET          PORDER       SORTED       IDENT
OWNER        SYSTEM
MEMBER       PRES
```

The SYSTEM record is a record type defined by the data base system.

3.5 Handling Groups

Since groups are not allowed in the data structure, a group must be defined in terms of "items". For example, "birthdate" in the President record must be defined in terms of month, day, year. Since another date group occurs in the President record (death date) more descriptive names must be chosen such as MONTHB, DAYB, and YEARB.

3.6 Handling Repeating Items

Repeating items are allowed by the DDL with some special declaration. Take the example of a repeating item as defined in the Presidential Data Structure which is "occupation" in the President record type. A maximum limit of how many times the item repeats must be specified in the DDL (in this case, 3). An item must be defined to maintain a count of the number of times the item repeats for any particular occurrence of the President record. In the DDL format, this information is presented as:

```
ITEM    NOOCC    INTEG    15
ITEM    OCCUP    CHAR     10     NOOCC     3
```

NOOCC is the item which maintains a count of the number of times OCCUP repeats. The "3" on the right specifies that, at most, a President can have three occupations listed.

Only one repeating item may be defined for a particular record type. Should there by more than one defined by the data structure, it should be thought of as a repeating group (which will be discussed next).

Last thing to be considered is that a repeating item is the last item to be defined (in the DDL format) for a particular record type.

3.7 Handling Repeating Groups

Any data defined in the data structure as a repeating group must be defined as a new record type in the DDL. For example, the repeating group, "marriage-date," in the President record must be defined as a new record type (call if MARRGE) and a 1:M relationship is formed between it and the PRES record (with PRES as the owner). The information in marriage-data can be presented by the following DDL:

```
RECORD    MARRGE
ITEM      WIFE       CHAR     10
ITEM      MONTHM     CHAR     10
ITEM      DAYM       INTEG    15
ITEM      YEARM      INTEG    31
ITEM      CHILDN     INTEG    15
SET       PRESME     FIFO
OWNER     PRES
MEMBER    MARRGE
```

3.8 Specifying Data Base Size

One DDL statement that does not directly describe the contents of the data base, but rather the size of it is NPAGES. The value associated with it (must be an integer) specifies the number of pages to be allocated for storing the contents of the data base. It's true value can only be estimated, but additions and deletions from the data base must be taken into account. Exceeding the NPAGES limit will result in a warning message.

3.9 Developing the DDL for the Presidential Example

From the original data structure for this example a new one can be constructed to show the added record types (due to repeating groups) and their relationships to the rest of the data structure.

Figure 3 presents the new data structure to be described by the DDL. Table 2 presents the names given to the records, sets and items defined in the DDL.

Finally, Figure 4 shows the complete DDL as derived from the conventions in sections 3.1 to 3.8.

Note that all items which contain character information (e.g., FSTNAM and IDENT) are declared to have a length, 10 characters, equal to that specified in PRESFILE. Though some occurrences of items carrying integer values (e.g., CAPTAL and AREA in the STATE record) have lengths of 15 bits and others of 31 bits, all have a fullword of storage assigned to them (31 bits of storage, 32 including the sign bit).

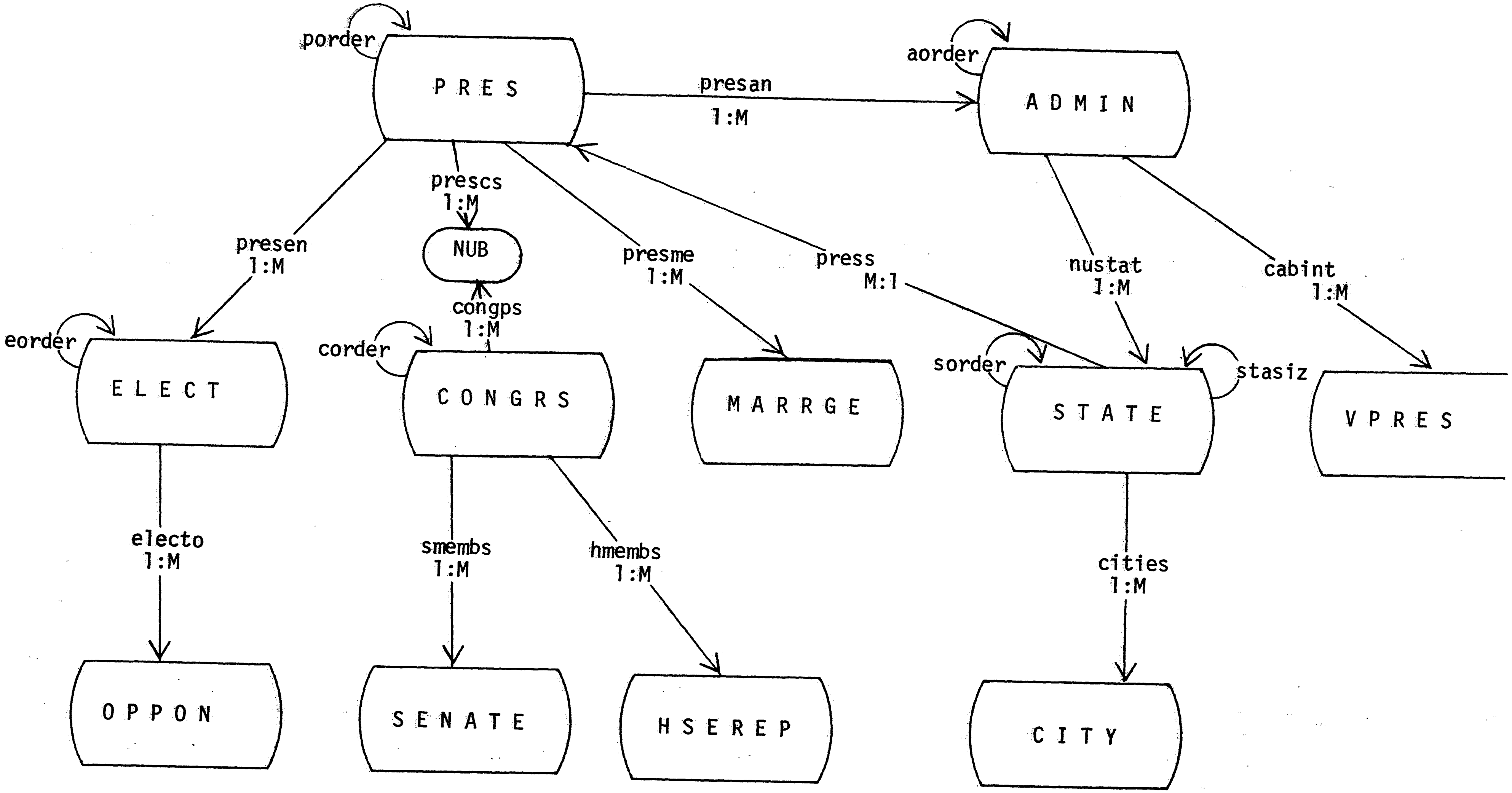


Figure 3

Record Relationships

Names used for Presidential Data Base
Records, Items and Sets

Record	Item	Set
PRES (president)	IDENT (identifier)	PRESEN (elections won)
	FSTNAM (first name)	PRESAN (administrations headed)
	INITAL (initial)	PRESCS (congresses headed)
	SURNAM (surname)	PRESS (state born in)
	MONTHB (month born)	PRESME (marriages)
	DAYB (day born)	
	YEARB (year born)	
	HEIGHT (height)	
	PARTY (political party)	
	COLEGE (college)	
	ANSTRY (ancestry)	
	RELIGN (religion)	
	MONTHD (month died)	
	DAYD (day died)	
	YEARD (year died)	
	CAUSE (cause of death)	
	FATHER (father's name)	
	MOTHER (mother's name)	
	NOOC (number of occupations)	
	OCCUP (name of occupation)	
MARRGE (marriage)	WIFE (wife's name)	
	MONTHM (month married)	
	DAYM (day married)	
	YEARM (year married)	
	CHILDN (number of children)	

Table 2

Record	Item	Set
ADMIN (administration	IDENT (identifier)	NUSTAT (new states added)
	MONTH (month started)	CABINT (vice presidents)
	DAY (day started)	
	YEAR (year started)	
VPRES (vice president)	FSTNAM (first name)	
	SURNAM (surname)	
ELECT (election)	IDENT (identifier)	ELECTO (opponents)
	YEAR (election year)	
	WINNER (winner)	
	PARTY (winning party)	
	VOTES (number of winning votes)	
OPPON (opponent)	NAME (opponent's name)	
	PARTY (opponent's party)	
	VOTES (opponent's number of votes)	
CONGRS (congress)	IDENT (identifier)	SMEMBS (senate members)
		HMEMBS (house members)
		CONGPS (headed by)
SENATE (senate)	PARTY (party name)	
	NUMBER (number of seats)	
HSEREP (house)	PARTY (party name)	
	NUMBER (number of seats)	

Table 2 (Continued)

Record	Item	Set
STATE (state)	IDENT ((identifier) NAME (state name) YEARAD (year admitted) CAPTAL (capital) AREA (area) ARANK (rank in area size) POP (polulation) PRANK (polulation rank) VOTES (electoral votes)	CITIES (cities)
CITY (city)	NAME (city name) POP (population)	

Ordering Relations

Record	Set	Ordered on
PRES	porder	IDENT (based on president's name)
ADMIN	aorder	IDENT (based on chronological order)
ELECT	eorder	IDENT (based on chronological order)
CONGRS	corder	IDENT (based on chronological order)
STATE	sorder	IDENT (based on state name)
STATE	stasiz	PRANK (based on polulation size)

Table 2 (Continued)

```
RECORD PRES
ITEM   IDENT  CHAR    10
ITEM   FSTNAM CHAR    10
ITEM   INITAL CHAR    10
ITEM   SURNAM CHAR    10
ITEM   MONTHB CHAR    10
ITEM   DAYB   INTEG   15
ITEM   YEARB  INTEG   31
ITEM   HEIGHT CHAR    10
ITEM   PARTY  CHAR    10
ITEM   COLEGE CHAR    10
ITEM   ANSTRY CHAR    10
ITEM   RELIGN CHAR    10
ITEM   MONTHD CHAR    10
ITEM   DAYD   INTEG   15
ITEM   YEARD  INTEG   31
ITEM   CAUSE  CHAR    10
ITEM   FATHER CHAR    10
ITEM   MOTHER CHAR    10
ITEM   NOOCC  INTEG   15
ITEM   OCCUP  CHAR    10 NOOCC      3
RECORD MARRGE
ITEM   WIFE   CHAR    10
ITEM   MONTHM CHAR    10
ITEM   DAYM   INTEG   15
ITEM   YEARM  INTEG   31
ITEM   CHILDN INTEG   15
RECORD ADMIN
ITEM   IDENT  CHAR    10
ITEM   MONTH  CHAR    10
ITEM   DAY    INTEG   15
ITEM   YEAR   INTEG   31
RECORD VPRES
ITEM   FSTNAM CHAR    10
ITEM   SURNAM CHAR    10
RECORD STATE
ITEM   IDENT  CHAR    10
ITEM   NAME   CHAR    10
ITEM   YEARAD INTEG   31
ITEM   CAPTAL CHAR    10
ITEM   AREA   INTEG   31
ITEM   ARANK  INTEG   15
ITEM   POP    INTEG   31
ITEM   PRANK  INTEG   15
ITEM   VOTES  INTEG   15
RECORD CITY
ITEM   NAME   CHAR    10
ITEM   POP    INTEG   31
```

Figure 4

The Presidential Data Base DDL

```
49      RECORD ELECT
50      ITEM   IDENT  CHAR    10
51      ITEM   YEAR   INTEG   31
52      ITEM   WINNER CHAR    10
53      ITEM   PARTY  CHAR    10
54      ITEM   VOTES  INTEG   15
55      RECORD OPPON
56      ITEM   NAME   CHAR    10
57      ITEM   PARTY  CHAR    10
58      ITEM   VOTES  INTEG   15
59      RECORD CONGRS
60      ITEM   IDENT  CHAR    10
61      RECORD SENATE
62      ITEM   PARTY  CHAR    10
63      ITEM   NUMBER INTEG   15
~~64      RECORD HSEREP~~
65      ITEM   PARTY  CHAR    10
66      ITEM   NUMBER INTEG   15
67      RECORD NUB
68      SET    PRESME FIFO
69      OWNER  PRES
70      MEMBER MARRGE
71      SET    PRESCS FIFO
72      OWNER  PRES
73      MEMBER NUB
74      SET    CONGPS IMMAT
75      OWNER  CONGRS
76      MEMBER NUB
77      SET    PRESEN SORTED     IDENT
78      OWNER  PRES
79      MEMBER ELECT
80      SET    PRESAN SORTED     IDENT
81      OWNER  PRES
82      MEMBER ADMIN
83      SET    NUSTAT SORTED     YEARAD
84      OWNER  ADMIN
85      MEMBER STATE
86      SET    CITIES SORTED     POP
87      OWNER  STATE
88      MEMBER CITY
89      SET    CABINT FIFO
90      OWNER  ADMIN
91      MEMBER VPRES
92      SET    SMEMBS SORTED     NUMBER
93      OWNER  CONGRS
94      MEMBER SENATE
95      SET    HMEMBS SORTED     NUMBER
96      OWNER  CONGRS
97      MEMBER HSEREP
```

Figure 4 (Continued)

```
  98     SET    STASIZ SORTED      PRANK
  99     OWNER  SYSTEM
 100     MEMBER STATE
 101     SET    PORDER SORTED      IDENT
 102     OWNER  SYSTEM
 103     MEMBER PRES
 104     SET    SORDER SORTED      IDENT
 105     OWNER  SYSTEM
 106     MEMBER STATE
 107     SET    CORDER SORTED      IDENT
 108     OWNER  SYSTEM
 109     MEMBER CONGRS
 110     SET    EORDER SORTED      IDENT
 111     OWNER  SYSTEM
 112     MEMBER ELECT
 113     SET    AORDER SORTED      IDENT
 114     OWNER  SYSTEM
 115     MEMBER ADMIN
 116     SET    ELECTO SORTED      VOTES
 117     OWNER  ELECT
 118     MEMBER OPPON
 119     SET    PRESS  IMMAT
 120     OWNER  STATE
 121     MEMBER PRES
 122     NPAGES                 20
```

Figure 4 (Continued)

4. CREATING AND INITIALIZING THE DATA BASE

After the DDL has been specified and stored in some way (either in a line file in the computer or on cards) it can be used to generate the Data Base Table File (DBTF). Once this is done the Data Base File (DBF) can be initialized (a preliminary action needed before any data can be stored in it).

4.1 Creation of DDL, DBTF and DBF files

As stated above, if the DDL for the data base is not punched on cards, it can be kept in a line file* in the computer. The DBTF and DBF files, however, must be sequential files*. If one DBMS is being developed it is a common convention to call the DDL file "DDL," the DBTF file, "DBTF" and DBF, "DBF." These names will be used in the following examples.

4.2 Generation of the DBTF*

First the DBTF file must be created:

```
$CREATE     DBTF     TYPE=SEQ
```

and then the following program should be run:

```
$RUN     SELW:DDLA                    5=DDL     8=DBTF
```

assuming that "DDL" is the name of the file containing the DDL for the DBMS. Two additional parameters may be used in conjunction with this program: 6=fdname and 7=fdname. 6 designates where the printed output from the program should be printed (it defaults to *SINK*). 7 designates file or device where block data generated from the program should be generated (it defaults to *SINK*).

4.3 Initialization of the DBF*

After the DBTF has successfully been generated and the DBF has been created by:

```
$CREATE     DBF     TYPE=SEQ
```

the data base file can be initialized by the following program:

```
$RUN     SELW:DBIN                    ~~1~~3=DBTF   2=DBF
```

Once this has been done the data base is ready to be accessed for storage and retrieval of data.

* The file types, commands, and programs used are particular to the Michigan Terminal System (MTS).

5. CODING CONVENTIONS FOR DATA BASE ACCESS PROGRAMS

Some conventions are common to any programs used to access the data base whether it is to store data, modify data stored, or retrieve data. (All examples of programs in this paper will be written in FORTRAN).

5.1 Common Areas and Block Data

All the RECORD, ITEM and SET names defined in the DDL (e.g., PRES, IDENT and PORDER in the Presidential Example) have to be represented as character strings (i.e., 'PRES   ', 'IDENT   ', and 'PORDER ', respectively) in order to be used by the Data Base Control System (DBCS) subroutines* for access programs. For example, a call to a DBCS routine could be stated as:

```
CALL  FMSK('PORDER   ',BUF,IERR)
```

An alternative to enclosing all DDL names in quotes is to define a constant to hold each character string:

```
INTEGER PORDER(2)
DATA PORDER/'PORD','ER  '/
```

(Note that the constant, PORDER must be of dimension '2' to hold six characters.** Now the same CALL statement can be written as

```
CALL  FMSK(PORDER,BUF,IERR)
```

which saves a considerable bit of coding since the constant will probably be used several times in a program.

A lot of coding effort can be saved by usage of the FORTRAN COMMON statements and BLOCK DATA facility. All of the constants' value assignments and type assignments (e.g., INTEGER) can be given by the BLOCK DATA subprogram and COMMON statements which car be used to relate these constants in BLOCK DATA to the main program and any subroutines.

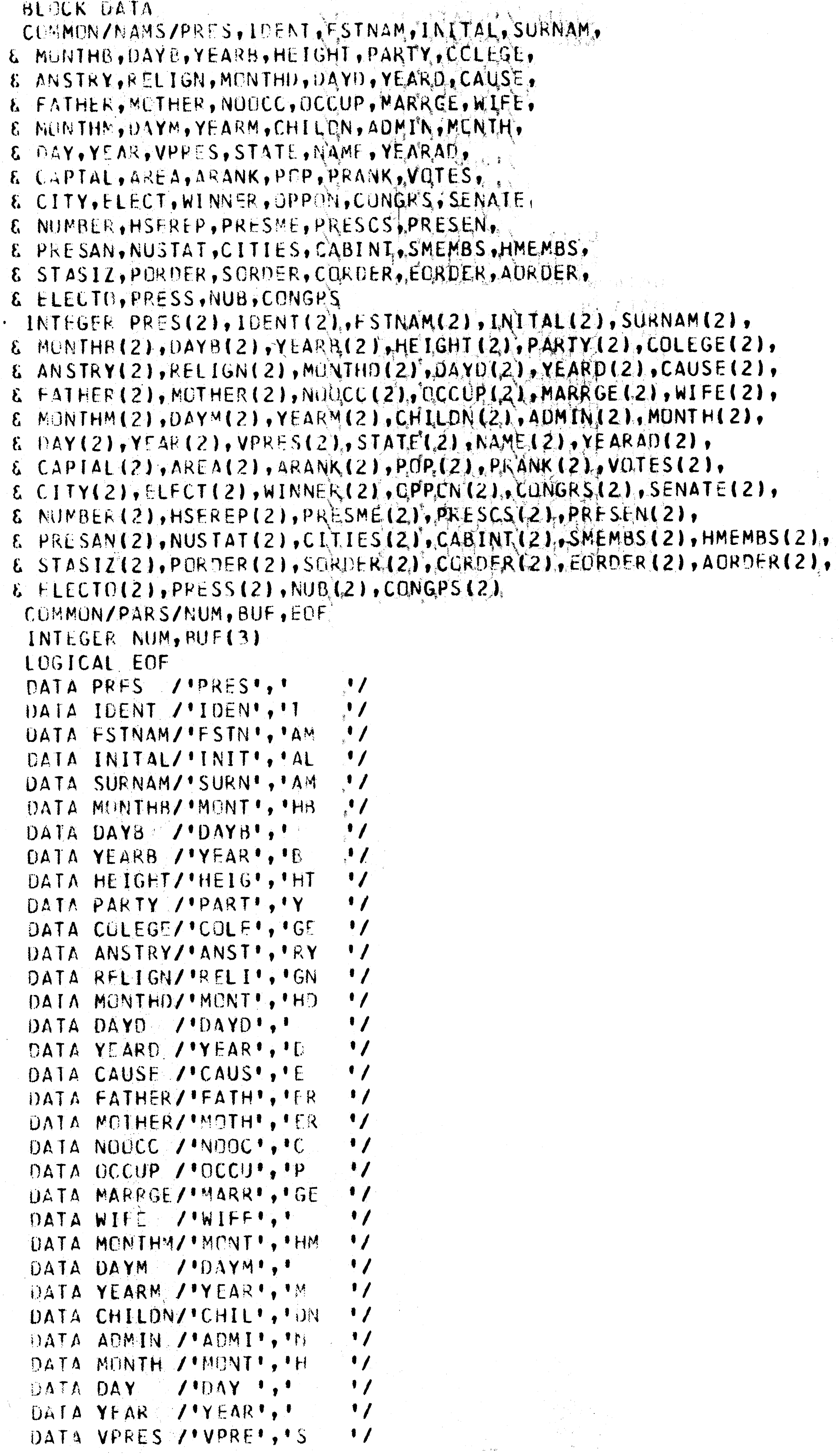
The information in the BLOCK DATA subprogram and COMMON areas are usually required for any subsequent programs and can be saved for this purpose also.

An example format of a BLOCK DATA subprogram is given in Figure 5. The COMMON area, NAMS, contains all the constant names. The second COMMON area, PARS, contains the buffer variables used in common with several programs and subroutines.

* Those subroutines described in ISDOS Working Paper No. 88

** If the DBMS is to be written to conform to ANSI standards, the DATA statement should specify Hollerith formats:

```
DATA PORDER/4HPORD,4HER  /
```



```
      BLOCK DATA
      COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
     & MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
     & ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
     & FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
     & MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
     & DAY,YEAR,VPRES,STATE,NAME,YEARAD,
     & CAPITAL,AREA,ARANK,POP,PRANK,VOTES,
     & CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
     & NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
     & PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
     & STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
     & ELECTO,PRESS,NUB,CONGPS
      INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
     & MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
     & ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
     & FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
     & MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
     & DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
     & CAPITAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
     & CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
     & NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
     & PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
     & STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
     & ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
      COMMON/PARS/NUM,BUF,EOF
      INTEGER NUM,BUF(3)
      LOGICAL EOF
      DATA PRES  /'PRES','    '/
      DATA IDENT /'IDEN','T   '/
      DATA FSTNAM/'FSTN','AM  '/
      DATA INITAL/'INIT','AL  '/
      DATA SURNAM/'SURN','AM  '/
      DATA MONTHB/'MONT','HB  '/
      DATA DAYB  /'DAYB','    '/
      DATA YEARB /'YEAR','B   '/
      DATA HEIGHT/'HEIG','HT  '/
      DATA PARTY /'PART','Y   '/
      DATA COLEGE/'COLE','GE  '/
      DATA ANSTRY/'ANST','RY  '/
      DATA RELIGN/'RELI','GN  '/
      DATA MONTHD/'MONT','HD  '/
      DATA DAYD  /'DAYD','    '/
      DATA YEARD /'YEAR','D   '/
      DATA CAUSE /'CAUS','E   '/
      DATA FATHER/'FATH','ER  '/
      DATA MOTHER/'MOTH','ER  '/
      DATA NOOCC /'NOOC','C   '/
      DATA OCCUP /'OCCU','P   '/
      DATA MARRGE/'MARR','GE  '/
      DATA WIFE  /'WIFE','    '/
      DATA MONTHM/'MONT','HM  '/
      DATA DAYM  /'DAYM','    '/
      DATA YEARM /'YEAR','M   '/
      DATA CHILDN/'CHIL','DN  '/
      DATA ADMIN /'ADMI','N   '/
      DATA MONTH /'MONT','H   '/
      DATA DAY   /'DAY ','    '/
      DATA YEAR  /'YEAR','    '/
      DATA VPRES /'VPRE','S   '/
```

Figure 5 Block data example

```
      DATA STATE /'STAT','E   '/
      DATA NAME  /'NAME','    '/
      DATA YEARAD/'YEAR','AD  '/
      DATA CAPTAL/'CAPT','AL  '/
      DATA AREA  /'AREA','    '/
      DATA ARANK /'ARAN','K   '/
      DATA POP   /'POP ','    '/
      DATA PRANK /'PRAN','K   '/
      DATA VOTES /'VOTE','S   '/
      DATA CITY  /'CITY','    '/
      DATA ELECT /'ELEC','T   '/
      DATA WINNER/'WINN','ER  '/
      DATA OPPON /'OPPO','N   '/
      DATA CONGRS/'CONG','RS  '/
      DATA SENATE/'SENA','TE  '/
      DATA NUMBER/'NUMB','ER  '/
      DATA HSEREP/'HSER','EP  '/
      DATA PRESME/'PRES','ME  '/
      DATA PRESCS/'PRES','CS  '/
      DATA PRESEN/'PRES','EN  '/
      DATA PRESAN/'PRES','AN  '/
      DATA NUSTAT/'NUST','AT  '/
      DATA CITIES/'CITI','ES  '/
      DATA CABINT/'CABI','NT  '/
      DATA SMEMBS/'SMEM','BS  '/
      DATA HMEMBS/'HMEM','BS  '/
      DATA STASIZ/'STAS','IZ  '/
      DATA PORDER/'PORD','ER  '/
      DATA SORDER/'SORD','ER  '/
      DATA CORDER/'CORD','ER  '/
      DATA EORDER/'EORD','ER  '/
      DATA AORDER/'AORD','ER  '/
      DATA ELECTO/'ELEC','TO  '/
      DATA PRESS /'PRES','S   '/
      DATA NUB   /'NUB ','    '/
      DATA CONGPS/'CONG','PS  '/
      END
OF FILE
```

Figure 5 Block data example

5.2 Buffer Variables

Buffer variables are variables used to temporary store data values as they pass to, or from, the data base. As specified in the COMMON area, PARS, NUM is a buffer variable used to hold all integer numerical data. BUF is a dimension 3 buffer variable which is used to hold all the ten character strings as read from PRESFILE. Note that in the Presidential Example all data is read in as ten characters and stored in BUF. Then the data is stored in the data base as a character string (ten characters) or converted to an integer and stored as a number. In DBMS, where many different sizes of data (measured in terms of fullwords), or types of data (logical, real, etc.) are to be read, a buffer variable must be defined for each different size and type. For example, if strings of 10 characters and 30 characters were to be manipulated the two buffers to handle these strings could be defined as:

```
INTEGER    BUF1(3),BUF2(8)
```

In any case, Table 1 given in Section 3, can be used as an aid in determining the dimension of the buffer variables to handle large character and bit strings. To define a buffer variable to hold double word size:

```
REAL* 8   RBUF
```

5.3 Access of the DBF and DBTF

Reference can be made to the Data Base File and Data Base Table File in a DBCS subroutine through assignment to logical IO units. For example, if the assignment of 2=DBF and 3=DBTF was made on the MTS $RUN command, the files could be referenced in the following manner:

```
CALL OPEN(2,3,100,IERR)
```

5.4 General Program Formats

Regardless of the purpose of the program accessing the data base a general format is inherent. The figure below presents the general format for main programs.

```
COMMON/NAMS/
      .
      .
      .
INTEGER ARRAY
      .              }  any data declarations particular
      .              }  to the main program
      .              }
CALL REPINT
CALL NEWREP(.FALSE.)
CALL OPEN(2,3,100,IERR)
                     }
                     }  program code
                     }
CALL CLOS(0,ARRAY,IERR)
END
```

REPINT and NEWREP perform initialization required before using the report subsystem subroutines (to be described later). OPEN specifies the DBF and DBTF files to be accessed by the program and CLOS closes these files properly. The figure below presents the general format for subroutines used in accessing the data base.

```
SUBROUTINE              routine-name
COMMON /NAMS/
       .
       .
       .
                      }  any data declarations particular
                      }  to the subroutine

                      }
                      }  code
                      }
RETURN
END
```

6. POPULATING THE DATA BASE

One major function of the DBMS is to populate the data base (i.e., store occurrences of each type of SET, RECORD and ITEM defined by DDL). There are several subroutines described in ISDOS Working Paper No. 88 and in the PSA software documentation which are helpful (if not necessary) in accomplishing this.

6.1 Reading Data from the Input File

Before any data can be stored in the data base it must first be read from some input file or device. The QROPEN, QREAD and QRCLOS routines can aid in reading the data. QROPEN specifies which file the data is to be read from, and the size of the logical record to be read by QREAD. For example,

```
CALL QROPEN(1,10,100)
```

specifies that the file used is assigned to Logical I/O unit 1 and the logical record size is ten characters within a block of size 100 (this obviously means that 10 records per block will be read as is the case of PRESFILE shown in Appendix A.)

QREAD reads in a logical record, stores the data and sets a switch should an end-of-file be encountered. For example,

```
CALL QREAD(10,BUF,EOF)
```

specifies that the first 10 characters (which in this case is all the characters) of the logical record be stored in the variable BUF. EOF is a logical variable which will be set to .TRUE. if an end-of-file is encountered. Since all data is read in a character format the subroutine CTOI can be used to convert character to integer. For example,

```
CALL CTOI(BUF,1,10,NUM)
```

will convert the character form of the number stored in BUF (starting in column 1 and which is ten characters long) to integer form and place the result in NUM. This is necessary when integer data is actually to be stored in the data base. If a data base "item" was defined to contain character information (in the DDL) then only character data can be stored in it. Likewise with integer or real numbers as data.

Finally QRCLOS closes the file designated in QROPEN. For example,

```
CALL QRCLOS
```

accomplishes this.

6.2 Storing Data

Now that data can be read by the access program in some form (character and integer are the forms concerning this application) it must be stored in the data base as an "item" for later retrieval. The following routines described in this section are considered the basic subset of subroutines (from ISDOS Working Paper No. 88) needed to populate a data base.*

FMSK	Used to find a particular member record occurrence in a particular set.
CR	Used to create record occurrences.
SFR	Used to store data in a particular "item" of a particular record.
AMS SRM SOM	Used to define relationships between a record occurrence and a particular set.
SMOVE	Used to store data in a "repeating item".

6.3 FMSK Find Member of a set based on Sort Key

This routine was used (in the Presidential example) to check that a record occurrence based on the record identifier (sort key) for a particular record type had not been previously stored in the data base. For example, if BUF contained the identifier "WASHINGT" the call

```
CALL FMSK(PORDER,BUF,IERR)
```

would check member record occurrences of the set PORDER to see that an occurrence has the sort key value contained in BUF. (The reader may note that this operation is consistent with the DDL specified for the Presidential example, i.e., PORDER is a defined set sorted on the item IDENT in the PRES record type and "WASHINGT" is the value IDENT will take on for the PRES record occurrence for George Washington.) If a record occurrence can not be found that has "WASHINGT" as an identifier, IERR will take on the value -1. Otherwise, the record occurrence found will be made the current member of the set PORDER. This makes it possible to store or retrieve data from the record occurrence. If IERR has the value -1, the record occurrence does not exist and a record occurrence must be created before any data can be stored in it.

6.4 CR Create a Record occurrence

This routine is used to create an occurrence of a particular record type (e.g., PRES) so that data may be stored in it and relationships to other record occurrences may be defined.

* The subroutine SMOVE description can be found in the PSA software documentation package.

For example, an occurrence of a PRES record type can be created by the call:

```
CALL CR(PRES,KEY,IERR)
```

The data base key value for this occurrence is returned in KEY and any error codes are returned in IERR.

6.5 SFR Set Field in the current Record

This routine is used to store data in a particular item (field) in a record occurrence. This would be used to store the character string "WASHINGT" contained in BUF in the item called IDENT in a PRES record occurrence. For example,

```
CALL SFR(IDENT,PRES,BUF,IERR)
```

would store the value of BUF in the item IDENT of the current PRES record occurrence. Any error code would be returned in IERR. It is important that BUF and IDENT are defined to hold the same type of data (i.e., INTEGER, CHARACTER, REAL, LOGICAL, etc.)

6.6 AMS Add a particular record occurrence as a Member of a Set

Now that IDENT has a value the record occurrence may be added to the set PORDER as a member. (This could not have been done previously because PORDER is sorted on IDENT values.) For example,

```
CALL AMS(PORDER,PRES,IERR)
```

adds the current PRES record occurrence to the set PORDER and makes this record occurrence the current member of the set. The record occurrence is added to the set according to any ordering criteria defined in the DDL (i.e., FIFO, SORTED, etc.).

At this point the particular record occurrence is the current member of the record type PRES and the current member record occurrence of the set PORDER.

6.7 SRM Set Record occurrence based on current Member of set

In the case where the member record occurrence was found, this routine can set the current member of a set to be the current record occurrence of a particular record type. For example,

```
CALL SRM(PORDER,IERR)
```

would set the current member record occurrence of the set PORDER to also be the current record occurrence of its particular record type (which in this case is the PRES record type).

At this point the particular record occurrence is the current record occurrence of the record type PRES and the current member record occurrence of the set PORDER.

6.8 SOM Set the current Owner of a set, based on the current Member of a set.

This routine is used to set up relationships between two particular record occurrences. Before a record occurrence is designated as a member of a set its owner must be defined. (In ordering relationships this is not necessary since there is only one owner record occurrences; SYSTEM). For example,

```
     CALL SOM(PRESME,PORDER,IERR)
```

makes the current member record of the set PORDER to be the current owner record of the set PRESME (marriage relationships). Once a marriage record (MARRGE) is created it can be related to a particular owner record by simply adding it to the PRESME set. For example,

```
     CALL AMS(PRESME,MARRGE,IERR)
```

designates that the current MARRGE record is a member of the set PRESME. Its related owner record is defined by the previously used SOM routine.

6.9 SMOVE String MOVE

This routine is used to store data in a repeating item. Using the example of the repeating item defined in the Presidential DDL:

```
     ITEM    NOOCC    INTEG    15
     ITEM    OCCUP    CHAR     10    NOOCC    3
```

Remember that the value of NOOCC is the number of occurrences of the repeating item OCCUP for any given occurrence of the PRES record. When NOOCC=3 there must be three occupations stored (each as a 10 character name). Also note that 3 is the maximum number of occurrences of the repeating item. Assuming that the number of occurrences of OCCUP (value of NOOCC) is supplied by PRESFILE before the values for OCCUP, (the format of PRESFILE shows this to be true) this data can be read and stored in the following manner.

```
     CALL QREAD(10,BUF,EOF)
     CALL CTOI(BUF,1,10,NUM)
     CALL SFR(NOOCC,PRES,NUM,IERR)
     IF (NUM .EQ. 0) GOTO 111
     PTR=1
     INC=NUM
     DO 11  I=1,INC
     CALL QREAD(10,BUF,EOF)
     CALL SMOVE(B,PTR,BUF,1,10)
11   PTR=PTR+10
     CALL SFR(OCCUP.PRES,B,IERR)
111             :
```

First the value to be given NOOCC is read, converted to integer representation and stored. A test is made to see that if any occurrences of OCCUP exist and branching is done accordingly. The repeating item OCCUP can be considered to be one large storage location (big enough for 3 values) and so a pointer is used (PTR) to specify where in the storage location the data will be stored. Figure 6 shows the format of this location.

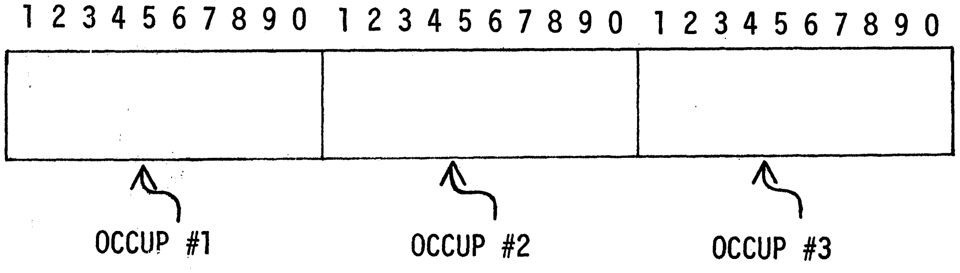


Figure 6

INC is used to contain the NOOCC value in the DO statement. (This is a convention used throughout the coding process, rather than using NUM. Since NUM might be used within a DO loop, its value can be modified accidentally.) Next, each occurrence of OCCUP is read and stored in the variable B (dimension 8, to hold thirty characters). Note that PTR, INC. and B are all declared as integer variables. SMOVE is then used to store each occurrance of OCCUP in B at its proper position (as designated by PTR). It is also specified that all ten (10) characters are to be stored starting at the first (1). Note that PTR will take on values of 1, 11 and 21. Finally, after all available occurrences of OCCUP have been stored in B the contents may be transferred to OCCUP in the data base via the SFR subroutine.

6.10 Code to Populate the Presidential Data Base

Now that some of the basic operations have been described they can all be related together to show how the data in PRESFILE can be stored in the Presidential data base.

A main program is used to identify the type of data that will be read from PRESFILE (i.e., one of the five types: PRES, ELECTION, ADMIN, CONGRESS or STATES data). After this is determined the appropriate subroutine is called that will handle storage of this data (i.e., CRPS, CREN, CRAN, CRCS or CRSS, respectively).

To generate the object code for this code:

```
$RUN *FTN PAR=SOURCE=BLOCFILE+ADMIN+CONGRS+ELECT+MAIN+PRES+STATE LOAD=CODE
```

BLOCFILE, ADMIN, etc. are the names of the files containing the subroutines, block data and main program code for the population operation, Table 3 presents relationships between the data in PRESFILE, the subprogram handling it and the file holding the source code for the subprogram.

Table 3

Date type in PRESFILE	Subprogram handling this data type	Source code file name
Data type identifiers	main program	MAIN
PRES data	CRPS subroutine	PRES
ELECTION data	CREN subroutine	ELECT
ADMIN data	CRAN subroutine	ADMIN
CONGRESS data	CRCS subroutine	CONGRS
STATES data	CRSS subroutine	STATE
---	BLOCK DATA	BLOCDATA

After the object code has been generated, the code can be linked up to the appropriate DBMS routines (e.g., QREAD, SFR, SMOVE, etc.) and executed by issuing:

```
$RUN  CODE+SELW:DBLIB     1=PRESFILE  2=DBF  3=DBTF
```

The rest of this section presents listings of the source code for the above subprograms.

```
        ***** MAIN PROGRAM *****

        * COMMON AREAS FOR MAIN *

COMMON/PARS/NUM,BUF,EOF
INTEGER NUM,BUF(3)
LOGICAL EOF

        * DEFINE VARIABLES TO BE USED IN THIS SUBPROGRAM *

INTEGER ARRAY

        * REPORT INITIALIZATION PROCEDURES *

CALL REPINT
CALL NEWREP(.FALSE.)

        * OPEN THE DATA BASE FILE AND DATA BASE TABLE FILE
          (LOGICAL I/O UNITS 2 AND 3 RESPECTIVELY) *

CALL OPEN(2,3,100,IERR)

        * OPEN PRESFILE(LOGICAL I/O UNIT 1) TO BE READ BY BLOCKED
          DATA I/O ROUTINES *

CALL QROPEN(1,10,100)

        * READ FIELD IN PRESFILE(SHOULD CONTAIN A DATA TYPE
          IDENTIFIER) *

CALL QREAD(10,BUF,EOF)

        * END PROCESSING SHOULD AN END-OF-FILE BE ENCOUNTERED *

IF (EOF) GOTO 999

        * TEST TO SEE IF THE IDENTIFIER SPECIFIES PRESIDENTIAL DATA *

J=ISCOMP(BUF,1,'PRES',1,4)
IF (J .NE. 0) GOTO 98

        * IF SO, CALL ROUTINE CRPS TO HANDLE THIS TYPE OF DATA *

CALL CRPS
GOTO 99

        * TEST TO SEE IF THE IDENTIFIER SPECIFIES ELECTION DATA *

J=ISCOMP(BUF,1,'ELECTION',1,8)
IF (J .NE. 0) GOTO 97

        * IF SO, CALL ROUTINE CREN TO HANDLE THIS TYPE OF DATA *

CALL CRET
GOTO 99

        * TEST TO SEE IF THE IDENTIFIER SPECIFIES ADMINISTRATION DATA *

7   J=ISCOMP(BUF,1,'ADMIN',1,5)
    IF (J .NE. 0) GOTO 96
```

```
C
C            * IF SO, CALL ROUTINE CRAN TO HANDLE THIS TYPE OF DATA *
C
      CALL CRAN
      GOTO 99
C
C            * TEST TO SEE OF THE IDENTIFIER SPECIFIES CONGRESS DATA *
C
 96   J=ISCOMP(BUF,1,'CONGRESS',1,8)
      IF (J .NE. 0) GOTO 95
C
C            * IF SO, CALL ROUTINE CRCS TO HANDLE THIS TYPE OF DATA *
C
      CALL CRCS
      GOTO 99
C
C            * TEST TO SEE IF THE IDENTIFIER SPECIFIES STATE DATA *
C
 95   J=ISCOMP(BUF,1,'STATES',1,6)
      IF (J .NE. 0) GOTO 94
C
C            * IF SO, CALL ROUTINE CRSS TO HANDLE THIS TYPE OF DATA *
C
      CALL CRSS
      GOTO 99
C
C            * ILLEGAL IDENTIFIER ENCOUNTERED, PRINT WARNING MESSAGE
C              AND STOP *
C
 94   CALL BUFBLD(1,1,31,31HWARNING-            RECORD-TYPE)
      CALL BUFBLD(10,1,10,BUF)
      CALL PBUF(2,0,.TRUE.)
C
C            * CLOSE PRESFILE *
C
 999  CALL QRCLOS
C
C            * CLOSE THE DATA BASE *
C
      CALL CLOS(0,ARRAY,IERR)
      END
```

```
SUBROUTINE CRPS

        ***** THIS ROUTINE HANDLES ALL DATA FOLLOWING A
              PRESIDENT DATA TYPE IDENTIFIER *****


        * COMMON AREAS FOR CRPS *

 COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
& MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
& ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
& FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
& MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
& DAY,YEAR,VPRES,STATE,NAME,YEARAD,
& CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
& CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
& NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
& PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
& STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
& ELECTO,PRESS,NUB,CONGPS
 INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
& MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
& ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
& FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
& MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
& DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
& CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
& CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
& NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
& PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
& STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
& ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
 COMMON/PARS/NUM,BUF,EOF
 INTEGER NUM,BUF(3)
 LOGICAL EOF

        * DEFINE VARIABLES TO BE USED IN THIS SUBPROGRAM *

 INTEGER B(8),PTR

        * READ THE PRESIDENT IDENTIFIER FROM PRESFILE *

 CALL QREAD(10,BUF,EOF)

        * TEST TO SEE IF THE PRESIDENT IS ALREADY IN THE
          DATA BASE *

 CALL FMSK(PORDER,BUF,IERR)
 IF (IERR .GT. -1) GOTO 110

        * IF SO, GO TO 110; IF NOT, CREATE A RECORD OCCURRENCE
          FOR HIM AND STORE THE IDENTIFIER *

 CALL CR(PRES,KEY,IERR)
 CALL SFR(IDENT,PRES,BUF,IERR)

        * ADD THIS RECORD OCCURRENCE TO THE SET 'PORDER' *

 CALL AMS(PORDER,PRES,IERR)
```

```
      GOTO 112
C
C             * MAKE THE RECORD THE CURRENT RECORD OCCURRENCE
C               OF THE RECORD TYPE 'PRES' *
C
 110  CALL SRM(PORDER,IERR)
C
C             * READ THE PRESIDENT'S LAST NAME AND STORE IT IN
C               'SURNAM' *
C
 112  CALL QREAD(10,BUF,EOF)
      CALL SFR(SURNAM,PRES,BUF,IERR)
C
C             * READ HIS FIRST NAME AND STORE IT IN 'FRSTNAM' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(FSTNAM,PRES,BUF,IERR)
C
C             * READ HIS MIDDLE INITIAL AND STORE IT IN 'INITAL' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(INITAL,PRES,BUF,IERR)
C
C             * READ HIS MONTH  OF BIRTH AND STORE IT IN 'MONTHB' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(MONTHB,PRES,BUF,IERR)
C
C             * READ HIS DAY OF BIRTH, CONVERT TO NUMBER, AND
C               STORE IN 'DAYB' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(DAYB,PRES,NUM,IERR)
C
C             * READ HIS YEAR OF BIRTH, CONVERT TO NUMBER, AND
C               STORE IN 'YEARB' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(YEARB,PRES,NUM,IERR)
C
C             * READ STATE NAME(AND STATE IDENTIFIER) BORN IN.
C               ONLY STATE IDENTIFIER NEED BE USED TO FORM THE
C               RELATIONSHIP BETWEEN 'PRES' AND 'STATE' RECORDS *
C
      CALL QREAD(10,BUF,EOF)
      CALL QREAD(10,BUF,EOF)
C
C             * TEST TO SEE IF THE STATE IS ALREADY IN THE DATA BASE *
C
      CALL FMSK(SORDER,BUF,IERR)
      IF (IERR .GT. -1) GOTO 101
C
C             * IF SO, GO TO 101: IF NOT, CREATE A RECORD OCCURRENCE
C               FOR IT AND STORE STATE IDENTIFIER IN IDENT *
C
      CALL CR(STATE,KEY,IERR)
      CALL SFR(IDENT,STATE,BUF,IERR)
C
```

```
        * ADD STATE RECORD OCCURRENCE TO SET 'SORDER' *

CALL AMS(SORDER,STATE,IERR)

        * FORM THE RELATIONSHIP(STATE OF BIRTH) BETWEEN
          PRESIDENT AND STATE *

CALL SOM(PRESS,SORDER,IERR)
CALL AMS(PRESS,PRES,IERR)

        * READ HIS HEIGHT AND STORE IN 'HEIGHT' *

CALL QREAD(10,BUF,EOF)
CALL SFR(HEIGHT,PRES,BUF,IERR)

        * READ HIS PARTY AND STORE IN 'PARTY' *

CALL QREAD(10,BUF,EOF)
CALL SFR(PARTY,PRES,BUF,IERR)

        * READ HIS COLLEGE ATTENDED AND STORE IN 'COLEGE' *

CALL QREAD(10,BUF,EOF)
CALL SFR(COLEGE,PRES,BUF,IERR)

        * READ HIS ANCESTRY AND STORE IN 'ANSTRY' *

CALL QREAD(10,BUF,EOF)
CALL SFR(ANSTRY,PRES,BUF,IERR)

        * READ HIS RELIGON AND STORE IN 'RELIGN' *

CALL QREAD(10,BUF,EOF)
CALL SFR(RELIGN,PRES,BUF,IERR)

        * READ NUMBER OF OCCUPATIONS, CONVERT TO NUMBER,
          AND STORE IN 'NOOCC' *

CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)
CALL SFR(NOOCC,PRES,NUM,IERR)

        * TEST TO SEE IF ANY OCCUPATION DATA SHOULDL BE STORED
          AS A REPEATING ITEM *

IF (NUM .EQ. 0) GOTO 111
PTR=1
INC=NUM

        * PACK THE OCCUPATIONS INTO A TEMPURARY BUFFER
          (BUF) BEFORE STORING IN DATA BASE *

DO 11 I=1,INC

        * READ AN OCCUPATION AND STORE IN 'BUF' *

CALL QREAD(10,BUF,EOF)
CALL SMOVE(B,PTR,BUF,1,10)
PTR=PTR+10
```

```
C               * STORE THE CONTENTS OF 'BUF' INTO 'OCCUP' *
C
      CALL SFR(OCCUP,PRES,B,IERR)
C
C             * READ MONTH DIED AND STORE IN 'MONTHD' *
C
 111  CALL QREAD(10,BUF,EOF)
      CALL SFR(MONTHD,PRES,BUF,IERR)
C
C             * READ DAY DIED, CONVERT TO NUMBER, AND STORE IN 'DAYD' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(DAYD,PRES,NUM,IERR)
C
C             * READ YEAR DIED, CONVERT TO NUMBER, AND STORE IN 'YEARD' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(YEARD,PRES,NUM,IERR)
C
C             * READ CAUSE OF DEATH AND STORE IN 'CAUSE' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(CAUSE,PRES,BUF,IERR)
C
C             * READ FATHER'S NAME AND STORE IN 'FATHER' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(FATHER,PRES,BUF,IERR)
C
C             * READ MOTHER'S NAME AND STORE IN 'MOTHER' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(MOTHER,PRES,BUF,IERR)
C
C             * READ NUMBER OF MARRIAGE DATA OCCURRENCES AND
C               CONVERT TO NUMBER *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
C
C             * SPECIFY RELATIONSHIP BETWEEN PRESIDENT AND
C               MARRIAGE DATA *
C
      CALL SOM(PRESME,PORDER,IERR)
      INC=NUM
C
C             * CREATE MARRIAGE RECORD AND STORE MARRIAGE DATA FOR
C               EACH MARRIAGE DATA OCCURRENCE *
C
      DO 12  I=1,INC
      CALL CR(MARRGE,KEY,IERR)
      CALL AMS(PRESME,MARRGE,IERR)
C
C             *READ NAME OF WIFE AND STORE IN 'WIFE' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(WIFE,MARRGE,BUF,IERR)
C
```

```
        * READ MONTH OF MARRIAGE AND STORE IN 'MONTHM' *

CALL QRFAD(10,BUF,EOF)
CALL SFR(MONTHM,MARRGE,BUF,IERR)

        * READ DAY OF MARRIAGE, CONVERT TO NUMBER, AND
          STORE IN 'DAYM' *

CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)
CALL SFR(DAYM,MARRGE,NUM,IERR)

        * READ YEAR OF MARRIAGE, CONVERT TO NUMBER, AND
          STORE IN 'YEARM' *

CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)
CALL SFR(YEARM,MARRGE,NUM,IERR)

        * READ NUMBER OF CHILDREN FROM MARRIAGE, CONVERT TO NUMBER,
          AND STORE IN 'CHILDN' *

CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)
CALL SFR(CHILDN,MARRGE,NUM,IERR)

        * READ NUMBER OF ELECTIONS WON AND CONVERT TO NUMBER *

CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)

        * IF PRESIDENT WON NO PRESIDENTIAL ELECTIONS, GO TO
          131; OTHERWISE, SPECIFY RELATIONSHIP BETWEEN
          PRESIDENT AND ELECTION DATA *

IF (NUM .EQ. 0) GOTO 131
CALL SOM(PRESEN,PORDER,IERR)
INC=NUM

        * PROCESS EACH OCCURRENCE OF ELECTION DATA *

DO 13 I=1,INC

        * READ ELECTION IDENTIFIER *

CALL QREAD(10,BUF,EOF)

        * TEST TO SEE IF ELECTION IS ALREADY STORED IN DATA BASE *

CALL FMSK(EORDER,BUF,IERR)
IF (IERR .GT. -1) GOTO 14

        * IF SO, GO TO 14; IF NOT, CREATE ELECTION RECORD
          AND STORE IDENTIFIER IN 'IDENT' *

CALL CR(ELECT,KEY,IERR)
CALL SFR(IDENT,ELECT,BUF,IERR)

        * ADD ELECTION RECORD TO SET 'EORDER' *
```

```
      CALL AMS(EORDER,ELECT,IERR)
      GOTO 13
C
C           * MAKE CURRENT MEMBER OF SET 'EORDER', CURRENT
C             RECORD OF TYPE 'ELECT' *
C
 14   CALL SRM(EORDER,IERR)
C
C           * FORM RELATIONSHIP BETWEEN PRESIDENT AND ELECTION
C             RECORD *
C
 13   CALL AMS(PRESEN,ELECT,IERR)
C
C           * READ NUMBER OF ADMINISTRATIONS AND CONVERT TO NUMBER *
C
 131  CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
C
C           * SPECIFY RELATIONSHIP BETWEEN  PRESIDENT AND
C             ADMINISTRATION DATA *
C
      CALL SOM(PRESAN,PORDER,IERR)
      INC=NUM
C
C           * PROCESS EACH OCCURRENCE OF ADMINISTRATION DATA *
C
      DO 17 I=1,INC
C
C           * READ ADMINISTRATION IDENTIFIER *
C
      CALL QREAD(10,BUF,EOF)
C
C           * TEST TO SEE IF ADMINISTRATION IS ALREADY STORED
C             IN DATA BASE *
C
      CALL FMSK(AORDER,BUF,IERR)
      IF (IERR .GT. -1) GOTO 16
C
C           *IF SO, GO TO 16; IF NOT, CREATE ADMINISTRATION RECORD
C             AND STORE IDENTIFIER IN 'IDENT' *
C
      CALL CR(ADMIN,KEY,IERR)
      CALL SFR(IDENT,ADMIN,BUF,IERR)
C
C           * ADD ADMINISTRATION RECORD TO SET 'AORDER' *
C
      CALL AMS(AORDER,ADMIN,IERR)
      GOTO 17
C
C           * MAKE CURRENT MEMBER OF SET 'AORDER', CURRENT
C             RECORD OF TYPE 'ADMIN' *
C
 16   CALL SRM(AORDER,IERR)
C
C           * FORM RELATIONSHIP BETWEEN PRESIDENT AND ADMINISTRATION
C             RECORD *
C
 17   CALL AMS(PRESAN,ADMIN,IERR)
C
C           * READ NUMBER OF CONGRESSES AND CONVERT TO NUMBER *
```

```
CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)
INC=NUM

        * PROCESS EACH OCCURRENCE OF CONGRESS DATA *

DO 18 I=1,INC

        * READ CONGRESS IDENTIFIER *

CALL QREAD(10,BUF,EOF)

        * TEST TO SEE IF CONGRESS IS ALREADY STORED IN DATA BASE *

CALL FMSK(CORDER,BUF,IERR)
IF (IERR .GT. -1) GOTO 19

        * IF SO, GO TO 19; IF NOT, CREATE CONGRESS RECORD AND
          STORE IDENTIFIER IN 'IDENT' *

CALL CR(CONGRS,KEY,IERR)
CALL SFR(IDENT,CONGRS,BUF,IERR)

        * ADD CONGRESS RECORD TO SET 'CORDER' *

CALL AMS(CORDER,CONGRS,IERR)

        * MAKE CURRENT MEMBER OF SET 'CORDER' , CURRENT
          RECORD OF TYPE 'CONGRS' *

CALL SOM(CONGRS,CORDER,IERR)

        * FORM RELATIONSHIP BETWEEN PRESIDENT AND CONGRESS
          RECORD THROUGH USE OF 'NUB' RECORD CONNECTION *

CALL SOM(PRESCS,PORDER,IERR)
CALL CR(NUB,KEY,IERR)
CALL AMS(CONGRS,NUB,IERR)
CALL AMS(PRESCS,NUB,IERR)
RETURN

        * PROCESSING OF DATA RELATED TO ONE PRESIDENT IS DONE *

END
```

```
      SUBROUTINE CRAN
C
C           ***** THIS ROUTINE HANDLES ALL DATA FOLLOWING AN
C             ADMINISTRATION IDENTIFIER *
C
C
C           * COMMON AREAS FOR CRAN *
C
      COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
     & MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
     & ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
     & FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
     & MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
     & DAY,YEAR,VPRES,STATE,NAME,YEARAD,
     & CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
     & CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
     & NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
     & PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
     & STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
     & ELECTO,PRESS,NUB,CONGPS
      INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
     & MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
     & ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
     & FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
     & MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
     & DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
     & CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
     & CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
     & NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
     & PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
     & STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
     & ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
      COMMON/PARS/NUM,BUF,EOF
      INTEGER NUM,BUF(3)
      LOGICAL EOF
C
C           * READ THE ADMINISTRATION IDENTIFIER FROM PRESFILE *
C
      CALL QREAD(10,BUF,EOF)
C
C           * TEST TO SEE IF THE ADMINISTRATION IS ALREADY IN
C             THE DATA BASE *
C
      CALL FMSK(AORDER,BUF,IERR)
      IF (IERR .GT. -1) GOTO 30
C
C           * IF SO, GO TO 30; IF NOT, CREATE A RECORD OCCURRENCE
C             FOR IT AND STORE THE IDENTIFIER IN 'IDENT' *
C
      CALL CR(ADMIN,KEY,IERR)
      CALL SFR(IDENT,ADMIN,BUF,IERR)
C
C           * ADD THIS RECORD TO THE SET 'AORDER' *
C
      CALL AMS(AORDER,ADMIN,IERR)
      GOTO 301
C
C           * MAKE THE RECORD THE CURRENT RECORD OCCURRENCE
C             OF THE RECORD TYPE 'ADMIN' *
C
```

```
0   CALL SRM(AORDER,IERR)

            * READ MONTH THE ADMINISTRATION STARTED AND STORE IN
              'MONTH' *

01  CALL QREAD(10,BUF,EOF)
    CALL SFR(MONTH,ADMIN,BUF,IERR)

            * READ DAY THE ADMINISTRATION STARTED, CONVERT TO
              NUMBER, AND STORE IN 'DAY' *

    CALL QREAD(10,BUF,EOF)
    CALL CTOI(BUF,1,10,NUM)
    CALL SFR(DAY,ADMIN,NUM,IERR)

            * READ YEAR THE ADMINISTRATION STARTED, CONVERT TO
              NUMBER, AND STORE IN 'YEAR' *

    CALL QREAD(10,BUF,EOF)
    CALL CTOI(BUF,1,10,NUM)
    CALL SFR(YEAR,ADMIN,NUM,IERR)

            * PRESIDENT NAME IS READ AND IGNORED( THIS INFORMATION
              IS ALSO GIVEN IN PRESIDENT DATA) *

    CALL QREAD(10,BUF,EOF)

            * READ NUMBER OF CABINET RECORDS AND CONVERT TO NUMBER *

    CALL QREAD(10,BUF,EOF)
    CALL CTOI(BUF,1,10,NUM)

            * IF NO CABINETS, GO TO 31, OTHERWISE, SPECIFY
              RELATIONSHIP BETWEEN ADMINISTRATION AND CABINET RECORD *

    IF (NUM .EQ. 0) GOTO 31
    CALL SOM(CABINT,AORDER,IERR)
    INC=NUM

            * PROCESS EACH OCCURRENCE OF CABINET DATA *

    DO 32 I=1,INC

            * CREATE CABINET RECORD OCCURRENCE *

    CALL CR(VPRES,KEY,IERR)

            * READ FIRST NAME OF VICE PRESIDENT AND STORE IN
              'FSTNAM' *

    CALL QREAD(10,BUF,EOF)
    CALL SFR(FSTNAM,VPRES,BUF,IERR)

            * READ LAST NAME OF VICE PRESIDENT AND STORE IN
              'SURNAM' *

    CALL QREAD(10,BUF,EOF)
    CALL SFR(SURNAM,VPRES,BUF,IERR)

            * FORM RELATIONSHIP BETWEEN ADMINISTRATION AND
```

```
C                CABINET RECORD *
C
 32   CALL AMS(CABINT,VPRES,IERR)
C
C             * READ NUMBER OF STATES ADMITTED DURING ADMINISTRATION
C               AND CONVERT TO NUMBER *
C
 31   CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
C
C             * IF NO STATES ADMITTED, STOP; OTHERWISE, SPECIFY
C               RELATIONSHIP BETWEEN ADMINISTRATION AND STATE RECORD *
C
      IF (NUM .EQ. 0) RETURN
      CALL SOM(NUSTAT,AORDER,IERR)
      INC=NUM
C
C             * PROCESS EACH OCCURRENCE OF STATE DATA *
C
      DO 34 I=1,INC
C
C             * READ STATE NAME(AND STATE IDENTIFIER) ADMITTED DURING
C               THE ADMINISTRATION. ONLY THE STATE IDENTIFIER NEED
C               BE USED TO FORM THE RELATIONSHIP BETWEEN 'ADMIN'
C               AND 'STATE' *
C
      CALL QREAD(10,BUF,EOF)
      CALL QREAD(10,BUF,EOF)
C
C             * TEST TO SEE IF THE STATE IS ALREADY IN THE DATA BASE *
C
      CALL FMSK(SORDER,BUF,IERR)
      IF (IERR .GT. -1) GOTO 35
C
C             * IF SO, GO TO 35; IF NOT, CREATE A RECORD OCCURRENCE
C               FOR IT AND STORE STATE IDENTIFIER IN 'IDENT' *
C
      CALL CR(STATE,KEY,IERR)
      CALL SFR(IDENT,STATE,BUF,IERR)
C
C             * ADD STATE RECORD OCCURRENCE TO SET 'SORDER' *
C
      CALL AMS(SORDER,STATE,IERR)
      GOTO 34
C
C             * MAKE THE RECORD THE CURRENT RECORD OCCURRENCE
C               OF THE RECORD TYPE 'STATE' *
C
 35   CALL SRM(SORDER,IERR)
C
C             * FORM RELATIONSHIP BETWEEN ADMINISTRATION AND
C               STATE RECORD *
C
 34   CALL AMS(NUSTAT,STATE,IERR)
      RETURN
C
C             * PROCESSING OF DATA RELATED TO ONE ADMINISTRATION IS DONE *
C
      END
```

```
SUBROUTINE CRET

        ***** THIS ROUTINE HANDLES ALL DATA FOLLOWING A
          ELECTION DATA IDENTIFIER *****


        * COMMON AREAS FOR CRET *

 COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
& MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
& ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
& FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
& MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
& DAY,YEAR,VPRES,STATE,NAME,YEARAD,
& CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
& CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
& NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
& PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
& STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
& ELECTO,PRESS,NUB,CONGPS
 INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
& MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
& ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
& FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
& MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
& DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
& CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
& CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
& NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
& PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
& STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
& ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
 COMMON/PARS/NUM,BUF,EOF
 INTEGER NUM,BUF(3)
 LOGICAL EOF

        * READ THE ELECTION IDENTIFIER FROM PRESFILE *

 CALL QREAD(10,BUF,EOF)

        * TEST TO SEE IF THE ELECTION IS ALREADY IN THE
          DATA BASE *

 CALL FMSK(EORDER,BUF,IERR)
 IF (IERR .GT. -1) GOTO 20

        * IF SO, GO TO 20; IF NOT, CREATE A RECORD OCCURRENCE
          FOR IT AND STORE THE IDENTIFIER *

 CALL CR(ELECT,KEY,IERR)
 CALL SFR(IDENT,ELECT,BUF,IERR)

        * ADD THIS RECORD OCCURRENCE TO THE SET 'EORDER' *

 CALL AMS(EORDER,ELECT,IERR)
 GOTO 201

        * MAKE THE RECORD THE CURRENT RECORD OCCURRENCE
          OF THE RECORD TYPE 'ELECT' *
 20   CALL SFM(ECRDER,IERR)
C
C            * READ THE YEAR OF THE ELECTION, CONVERT TO NUMBER,
C              AND STORE IN 'YEAR' *
C
 201  CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(YEAR,ELECT,NUM,IERR)
C
C            * READ THE WINNER OF THE ELECTION AND STORE IN 'WINNER' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(WINNER,ELECT,BUF,IERR)
C
C            * READ THE WINNING PARTY AND STORE IN 'PARTY' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(PARTY,ELECT,BUF,IERR)
C
C            * READ THE NUMBER OF WINNING VOTES, CONVERT TO  NUMBER,
C              AND STORE IN 'VOTES' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(VOTES,ELECT,NUM,IERR)
C
C            * READ NUMBER OF OPPONENTS AND CONVERT TO NUMBER *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
C
C            * SPECIFY RELATIONSHIP BETWEEN ELECTION AND OPPONENT
C              RECORD *
C
      CALL SON(ELECTC,ECRDER,IERR)
      INC=NUM
C
C            * PROCESS EACH OCCURRENCE OF OPPONENT DATA *
C
      DO 21 I=1,INC
C
C            * CREATE AN OCCURRENCE OF AN OPPONENT RECORD AND
C              FORM RELATIONSHIP BETWEEN OPPONENT AND ELECTION
C              RECORD *
C
      CALL CR(OPPON,KEY,IERR)
      CALL AMS(ELECTC,OPPON,IERR)
C
C            * READ THE NAME OF OPPONENT AND STORE IN NAME *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(NAME,OPPON,BUF,IERR)
C
C            * READ OPPONENT'S PARTY AND STORE IN 'PARTY' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(PARTY,OPPON,BUF,IERR)
C
C            * READ NUMBER OF OPPONENT'S VOTES, CONVERT TO NUMBER,
C              AND STORE IN 'VOTES' *
```

```
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
21    CALL SFR(VOTES,OPPON,NUM,IERR)
      RETURN

              * PROCESSING OF DATA RELATED TO ONE ELECTION IS DONE *

      END
      SUBROUTINE CRCS
C
C              ***** THIS ROUTINE HANDLES ALL DATA FOLLOWING A
C                CONGRESS DATA IDENTIFIER *****
C
C
C              * COMMON AREAS FOR CRCS *
C
      COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
     & MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
     & ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
     & FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
     & MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
     & DAY,YEAR,VPRES,STATE,NAME,YEARAD,
     & CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
     & CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
     & NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
     & PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
     & STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
     & ELECTO,PRESS,NUB,CONGPS
      INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
     & MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
     & ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
     & FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
     & MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
     & DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
     & CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
     & CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
     & NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
     & PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
     & STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
     & ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
      COMMON/PARS/NUM,BUF,EOF
      INTEGER NUM,BUF(3)
      LOGICAL EOF
C
C              * READ THE CONGRESS IDENTIFIER FROM PRESFILE *
C
      CALL QREAD(10,BUF,EOF)
C
C              * TEST TO SEE IF THE ELECTION IS ALREADY IN THE
C                DATA BASE *
C
      CALL FMSK(CORDER,BUF,IERR)
      IF (IERR .GT. -1) GOTO 401
C
C              * IF SO, GO TO 401; IF NOT, CREATE A RECORD OCCURRENCE
C                FOR IT AND STORE THE IDENTIFIER *
C
      CALL CR(CONGRS,KEY,IERR)
      CALL SFR(IDENT,CONGRS,BUF,IERR)
C
C              * ADD THIS RECORD OCCURRENCE TO THE SET 'CORDER' *
C
      CALL AMS(CORDER,CONGRS,IERR)
      GOTO 40
C
C              * MAKE THE RECORD THE CURRENT RECORD OCCURRENCE
C                OF THE RECORD TYPE 'CONGRS' *
C
```

```
1   CALL SRM(CORDER,IERR)

           * READ THE NUMBER OF SENATE PARTIES AND CONVERT TO NUMBER *

    CALL QREAD(10,BUF,EOF)
    CALL CTOI(BUF,1,10,NUM)

           * SPECIFY RELATIONSHIP BETWEEN CONGRESS AND SENATE RECORD *

    CALL SOM(SMEMBS,CORDER,IERR)
    INC=NUM

           * PROCESS EACH OCCURRENCE OF SENATE DATA *

    DO 41 I=1,INC

           * CREATE AN OCCURRENCE OF A SENATE RECORD  *

    CALL CR(SENATE,KEY,IERR)

           * READ THE SENATE PARTY AND STORE IN 'PARTY' *

    CALL QREAD(10,BUF,EOF)
    CALL SFR(PARTY,SENATE,BUF,IERR)

           * READ THE NUMBER OF MEMBERS, CONVERT TO NUMBER,
             AND STORE IN 'NUMBER' *

    CALL QREAD(10,BUF,EOF)
    CALL CTOI(BUF,1,10,NUM)
    CALL SFR(NUMBER,SENATE,NUM,IERR)

           * FORM RELATIONSHIP BETWEEN CONGRESS AND SENATE RECORD *

l   CALL AMS(SMEMBS,SENATE,IERR)

           * READ THE NUMBER OF HOUSE PARTIES, CONVERT TO NUMBER *

    CALL QREAD(10,BUF,EOF)
    CALL CTOI(BUF,1,10,NUM)

           * SPECIFY RELATIONSHIP BETWEEN CONGRESS AND HOUSE RECORD *

    CALL SOM(HMEMBS,CORDER,IERR)
    INC=NUM

           * CREATE AN OCCURRENCE OF A HOUSE RECORD *

    DO 42 I=1,INC
    CALL CR(HSEREP,KEY,IERR)

           * READ THE HOUSE PARTY AND STORE IN 'PARTY' *

    CALL QREAD(10,BUF,EOF)
    CALL SFR(PARTY,HSEREP,BUF,IERR)

           * READ THE NUMBER OF MEMBERS, CONVERT TO NUMBER,
             AND STORE IN 'NUMBER' *

    CALL QREAD(10,BUF,EOF)
```

```
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(NUMBER,HSEREP,NUM,IERR)
C
C             * FORM RELATIONSHIP BETWEEN CONGRESS AND HOUSE RECORD *
C
 42   CALL AMS(HMEMBS,HSEREP,IERR)
      RETURN
C
C             * PROCESSING OF DATA RELATED TO ONE CONGRESS IS DONE *
C
      END
 SUBROUTINE CRSS

        ***** THIS ROUTINE HANDLES ALL DATA FOLLOWING A
          STATE DATA IDENTIFIER *****


        * COMMON AREAS FOR CRSS *

 COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
& MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
& ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
& FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
& MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
& DAY,YEAR,VPRES,STATE,NAME,YEARAD,
& CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
& CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
& NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
& PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
& STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
& ELECTO,PRESS,NUB,CONGPS
 INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
& MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
& ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
& FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
& MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
& DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
& CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
& CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
& NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
& PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
& STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
& ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
 COMMON/PARS/NUM,BUF,EOF
 INTEGER NUM,BUF(3)
 LOGICAL EOF

        * READ THE STATE IDENTIFIER FROM PRESFILE *

 CALL QREAD(10,BUF,EOF)

        * TEST TO SEE IF THE STATE IS ALREADY IN THE
          DATA BASE *

 CALL FMSK(SORDER,BUF,IERR)
 IF (IERR .GT. -1) GOTO 50

        * IF SO, GO TO 50; IF NOT, CREATE A RECORD OCCURRENCE
          FOR IT AND STORE THE IDENTIFIER *

 CALL CR(STATE,KEY,IERR)
 CALL SFR(IDENT,STATE,BUF,IERR)

        * ADD THIS RECORD OCCURRENCE TO THE SET 'SORDER' *

 CALL AMS(SORDER,STATE,IERR)
 GOTO 51

        * MAKE THE RECORD THE CURRENT RECORD OCCURRENCE
          OF THE RECORD TYPE 'STATE' *
 50   CALL SRM(SORDER,IERR)
C
C           * READ THE NAME OF THE STATE AND STORE IN 'NAME' *
C
 51   CALL QREAD(10,BUF,EOF)
      CALL SFR(NAME,STATE,BUF,IERR)
C
C           * READ THE YEAR ADMITTED TO THE UNION, CONVERT TO NUMBER,
C             AND STORE IN 'YEARAD' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(YEARAD,STATE,NUM,IERR)
C
C           * READ THE CAPITAL CITY AND STORE IN 'CAPTAL' *
C
      CALL QREAD(10,BUF,EOF)
      CALL SFR(CAPTAL,STATE,BUF,IERR)
C
C           * READ THE AREA OF THE STATE, CONVERT TO NUMBER,
C             AND STORE IN 'AREA' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(AREA,STATE,NUM,IERR)
C
C           * READ THE RANK OF THE STATE (IN TERMS OF AREA SIZE),
C             CONVERT TO NUMBER, AND STORE IN 'ARANK' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(ARANK,STATE,NUM,IERR)
C
C           * READ THE POPULATION OF THE STATE, CONVERT TO NUMBER,
C             AND STORE IN 'POP' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(POP,STATE,NUM,IERR)
C
C           * READ THE RANK OF THE STATE (IN TERMS OF POPULATION SIZE),
C             CONVERT TO NUMBER, AND STORE IN 'PRANK' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(PRANK,STATE,NUM,IERR)
C
C           * ADD THIS RECORD OCCURRENCE TO THE SET 'STASIZ' *)
C
      CALL AMS(STASIZ,STATE,IERR)
C
C           * READ THE NUMBER OF ELECTORAL VOTES FOR THE STATE,
C             CONVERT TO NUMBER, AND STORE IN 'VOTES' *
C
      CALL QREAD(10,BUF,EOF)
      CALL CTOI(BUF,1,10,NUM)
      CALL SFR(VOTES,STATE,NUM,IERR)
C
C           * READ NUMBER OF MAJOR CITIES AND CONVERT TO NUMBER *
C
```

```
CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)

       * IF NO MAJOR CITIES, STOP ; OTHERWISE SPECIFY
         RELATIONSHIP BETWEEN STATE AND CITY RECORD *

IF (NUM .EQ. 0) RETURN
CALL SOM(CITIES,SORDER,IERR)
INC=NUM

       * PROCESS EACH OCCURRENCE OF CITY DATA *

DO 53 I=1,INC

       * CREATE AN OCCURRENCE OF A CITY RECORD *

CALL CR(CITY,KEY,IERR)

       * READ NAME OF CITY AND STORE IN 'NAME' *

CALL QREAD(10,BUF,EOF)
CALL SFR(NAME,CITY,BUF,IERR)

       * READ POPULATION OF CITY AND STOR IN 'POP' *

CALL QREAD(10,BUF,EOF)
CALL CTOI(BUF,1,10,NUM)
CALL SFR(POP,CITY,NUM,IERR)

       * FORM RELATIONSHIP BETWEEN STATE AND CITY RECORD *

CALL AMS(CITIES,CITY,IERR)
RETURN

       * PROCESSING OF DATA RELATED TO ONE STATE IS DONE *

END
```

7. GENERATING REPORTS FROM THE DATA BASE

Once data has been successfully entered into the data base there are several routines (described in ISDOS Working Paper No. 88 and the PSA software documentation package) which can be used to aid in the retrieval process and in formatting the retrieved data.

7.1 Retrieving Data

There are several routines available to the user (of which just a few are described here) to aid in the retrieval of data in a random fashion or via some ordered manner.

FNM FFM	Used to locate member record occurrences for a given owner record occurrence.
SOM SMM	Used to retrieve data occurrences based on some defined relationship.
GFM GFO	Used to retrieve data from a particular "item" occurrence.
SMOVE	Used to retrieve data from a repeating item.

7.2 FFM Find First Member record occurrence

This routine is used to find the first member record occurrence for the current owner record in a given set. For example, if a listing of all President names were to be retrieved and printed the call

```
CALL   FFM (PORDER, IERR)
```

finds the first member record occurrence of the set PORDER. (Looking back, notice that the DDL designated the owner record for this set to be SYSTEM and member record occurrences to be of the type PRES.) At this point the first member occurrence has been located or an error code has been returned in IERR. Data can now be taken from the record occurrence (if no error was encountered).

7.3 FNM Find Next Member record occurrence

This routine can be used to find the first member record occurrence (like FFM) if there is no current member record occurrence, or can be used to find any subsequent member record occurrences in the order specified in the DDL. For example,

```
CALL   FNM (PORDER, IERR)
```

will find the next PRES record occurrence, based on the value of the IDENT item. (Remember that the set PORDER is ordered on the IDENT items within the PRES record occurrences.) A return code value of -1 will be given to IERR should an end-of-file condition be encountered.

7.4 SOM Set the current Owner of a set, based on the current Member of a set.

This routine is used (as in SMM) to locate and retrieve data based on specific relationships between the record types in the data base. For example, for a PRES record occurrence (found by FFM or FNM) it may be desirable to find those ELECT record occurrences related to it via the PRESEN relationship.

```
CALL   SOM (PRESEN, PORDER, IERR)
```

specifies that the current member record of set PORDER (which would be a PRES record type) is to be made the current owner record of the set PRESEN (whose members are ELECT record types). Once this has been set, the first member record occurrence of the set PRESEN is available for any retrieval procedures. The next ELECT member record occurrence for the particular PRES record occurrence can be located by the FNM routine:

```
CALL   FNM (PRESEN, IERR)
```

7.5 SMM Set the current Member of a set, based on the current Member of a set.

This routine can be used to retrieve data for a record occurrence which is a member of two different types of sets. For example, all PRES record occurrences are members of the set PORDER (ordering of all Presidents by identifier) as well as the set PRESS (relationship between Presidents and State born in). So, given that a particular PRES record occurrence is available via FFM or FFM, the STATE record occurrence the President is related to can be found by the call

```
CALL   SMM (PRESS, PORDER, IERR)
```

where the current member of the set PORDER also now becomes the current member of the set PRESS. State data can now be extracted.

7.6 GFM Get Field from Member record occurrence

Once the appropriate record occurrence has been found by the above routines, GFM is the routine used to actually retrieve the data from the record. For example, to retrieve the first name of a President the call

```
CALL   GFM (FSTNAM, PORDER, BUF, IERR)
```

would be used. This retrieves the data from the "item" FSTNAM in the current member record occurrence of the set PORDER (which means this is a PRES record occurrence) and places the data in BUF. BUF must be of the same data type (i.e. integer, character, decimal, etc.) as the "item" and large enough to hold its value.

7.7 GFO Get Field from Owner record occurrence

This routine is basically the same as GFM, but retrieves data from an owner record occurrence rather than a member. Looking back at the SMM routine, a STATE record occurrence was found as a result of

```
CALL    SMM (PRESS, PORDER, IERR)
```

which also made the STATE record occurrence the current owner of the set PRESS. GFO allows data from the STATE record to be retrieved.

```
CALL    GFO (NAME, PRESS, BUF, IERR)
```

retrieves the data in the "item" NAME from the current owner record occurrence of the set PRESS, and puts the data into BUF.

7.8 Report Routines

As the data is being retrieved it has to be formatted in some manner to be presented as a report. The following routines aid in this process.

HEDING	Used to print a heading for the report
BUFBLD NINBUF	Used to store data in the output buffer to be printed
PBUF	Used to print out contents of the output buffer

7.9 HEDING (Heading)

This subroutine allows the programmer to specify a title for a report. If the report is more than one page long, the title will be presented at the top of each new page. For example, by specifying

```
CALL    HEDING(31,31H** PRESIDENT/ELECTION REPORT **,0,3)
```

the title "** PRESIDENT/ELECTION REPORT **" will be printed at the top of each page of the report. The first parameter value, 31, designates that the title is 31 characters long. The zero (0) parameter value above specifies that the title should be centered. (Any positive integers in place of this parameter value specifies the column where the title begins.) Finally, the last parameter value, 3, designates the number of spaces to skip before printing out the contents of the report.

7.10 BUFBLD (Buffer Build)

This routine allows the storage of character data into an output buffer which when printed, becomes part of the report. For example,

```
CALL  BUFBLD(1,1,14,14HPRESIDENT NAME)
```

stores the character string "PRESIDENT NAME" into the first position (1) in the output buffer (as designated by the first paramemter). The third parameter (here 14) specifies the number of characters in the string. Taking another example,

```
CALL   BUFBLD(28,1,3,BUF)
```

this specifies that the contents of the variable (a character array), BUF, should be stored in the output buffer, starting at the 28th position in the buffer. The second parameter specifies an index into BUF where the character string starts. (In this case the character data of interest starts in the first (1) position.) 3 specifies the length of the character string to be stored. Note that if BUF contained a twelve character string, the last 9 characters would be ignored while length is specified to be 3.

7.11 NINBUF (Numeric Integer into Buffer)

This routine is used much in the same way as BUFBLD but, instead places numeric data (rather than character) into the output buffer. For example,

```
CALL   NINBUF(NUM,21,4)
```

converts the numeric data in the variable NUM into character format and stores this starting at position 21 in the output buffer. The last parameter specifies the length of the number being stored. The number is assumed to be an integer.

7.12 PBUF (Print Buffer)

This routine prints out the contents of the output buffer set up by the BUFBLD and NINBUF routines. By specifying

```
PBUF(0,1,.TRUE.)
```

The output buffer is printed, a line is skipped after the line is printed (1), and the buffer is blanked out (.TRUE.). The zero (0) parameter designates that no lines should be skipped before printing the buffer. (Of course, this can have any positive integer value.) If the buffer is not blanked out (.FALSE.) data stored in the buffer will be overlayed over the previous data.

7.13 Code to Generate Reports from the Presidential Data Base

Now that the basic routines needed to generate reports from a data base have been described, examples of how these routines can be used to actually generate reports will be given in this section.

The common areas and block data used in the population of the data base can be used in these report programs to reduce effort in defining variables and constants.

Assuming the source code for the report program to be in TESTPROG, the object code can be put into TEST when

```
     $RUN *FTN  PAR=SOURCE=TESTPROG  LOAD=TEST
```

is given. Assuming BLOCKFILE contains the object version of the block data subprogram for the data base the report program can be executed by the command:

```
     $RUN  TEST+BLOCKFILE+SELW:DBLIB   2=DBF  3=DBTF
```

where DBF is the data base file and DBTF is the data base table file.

The remainder of this section presents descriptions and listings of programs that generate reports from the Presidential data base. Appendix B presents the reports generated from these programs.

TEST PROGRAM #1

This program obtains an alphabetical listing of all presidents in the data base. It prints out first and last name and middle initial if there is one. For each president, it also finds all elections related to that president that won him the presidency and the date of the election, and at least one of the opponents he beat. In some cases a particular opponent is not specified but rather that a number of opponents were defeated. For those presidents with no election information, nothing is given.

```
C            **** PRESIDENT/ELECTION REPORT PROGRAM ****
C
C            * THIS PROGRAM RETRIEVES AND PRINTS ELECTION INFORMATION
C              RELATIVE TO EACH PRESIDENT *
C
C            * COMMON AREAS FOR THE PROGRAM *
C
      COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
     & MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
     & ANSTRY,RELIGN,MCNTHD,DAYD,YEARD,CAUSE,
     & FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
     & MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
     & DAY,YEAR,VPRES,STATE,NAME,YEARAD,
     & CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
     & CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
     & NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
     & PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
     & STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
     & ELECTO,PRESS,NUB,CONGPS
      INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
     & MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
     & ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
     & FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
     & MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
     & DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
     & CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
     & CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
     & NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
     & PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
     & STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
     & ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
      COMMON/PARS/NUM,BUF,EOF
      INTEGER NUM,BUF(3)
      LOGICAL EOF
C
C            * REPORT INITIALIZATION PROCEDURES *
C
      CALL REPINT
      CALL NEWREP(.FALSE.)
C
C            * SPECIFY THE TITLE TO BE PRINTED AT TOP OF EACH PAGE OF
C              THE REPORT *
C
      CALL HEDING(31,31H** PRESIDENT/ELECTION REPORT **,0,3)
C
C            * OPEN THE DATA BASE FILE AND THE DATA BASE TABLE FILE
C              ( LOGICAL I/O UNITS 2 AND 3 RESPECTIVELY ) *
C
      CALL OPEN(2,3,100,IERR)
C
C            * FIND PRESIDENT RECORD OCCURRENCE. IF NONE CAN BE FOUND,
C              STOP *
C
 1    CALL FNM(PORDER,IERR)
      IF (IERR .EQ. -1) GOTO 100
C
C            * PLACE COMMENT INTO OUTPUT BUFFER *
C
      CALL BUFBLD(1,1,14,14HPRESIDENT NAME)
C
```

```
         * RETREIVE PRESIDENT'S FIRST NAME, MIDDLE INITIAL AND
           SURNAME  AND PLACE THEM INTO THE OUTPUT BUFFER *

    CALL GFM(FSTNAM,PORCER,BUF,IERR)
    CALL BUFBLD(16,1,10,BUF)
    CALL GFM(INITAL,PCRDER,BUF,IERR)
    CALL BUFBLD(28,1,3,BUF)
    CALL GFM(SURNAM,PORCER,BUF,IERR)
    CALL BUFBLD(31,1,10,BUF)

         * PRINT OUT BUFFER WITH PRESIDENT'S NAME *

    CALL PBUF(2,0,.TRUE.)

         * TEST TO SEE IF THE PRESIDENT WON ANY ELECTIONS FOR THE
           PRESIDENCY AND IF NOT, GO PROCESS THE NEXT PRESIDENT *

    CALL SOM(PRESEN,PCRDER,IERR)
25  IF (IERR .EQ. -1) GOTO 1

         * PLACE COMMENT INTO OUTPUT BUFFER *

    CALL BUFBLD(4,1,29,29HWON ELECTION OF       AGAINST)

         * RETRIEVE YEAR PRESIDENT WON ELECTION AND PLACE IN OUTPUT
           BUFFER *

    CALL GFM(YEAR,PRESEN,NUM,IERR)
    CALL NINBUF(NUM,21,4)

         * FIND OPPONENT PRESIDENT BEAT IN THIS ELECTION AND PLACE
           INTO OUTPUT BUFFER *

    CALL SOM(ELECTO,PRESEN,IERR)
    CALL GFM(NAME,ELECTC,BUF,IERR)
    CALL BUFBLD(34,1,10,BUF)

         * PRINT CUT CUTPUT BUFFER WITH ELECTION INFORMATION *

    CALL PBUF(0,0,.TRUE.)

         * CUNTINUE TO PROCESS ANY REMAINING ELECTION RECORDS
           RELATED TC PRESIDENT *

    CALL FNM(PRESEN,IERR)
    GOTO 25

         * CLOSE THE DATA BASE *

100 CALL CLOS(0,ARRAY,IERR)
    END
```

TEST PROGRAM #2

This program obtains an alphabetical listing of all states in the data base. For each state it prints out state name and year admitted to the union. It also finds any city records related to that state and prints out city name as well as city population.

```
C          **** STATE/CITY REPORT PROGRAM ****
C
C          * THIS PROGRAM RETRIEVES CITY INFORMATION AND HISTORICAL
C            INFORMATION RELATIVE TO EACH STATE *
C
C
C          * COMMON AREAS FOR THE PROGRAM *
C
      COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
     & MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
     & ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
     & FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
     & MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
     & DAY,YEAR,VPRES,STATE,NAME,YEARAD,
     & CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
     & CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
     & NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
     & PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
     & STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
     & ELECTO,PRESS,NUB,CONGPS
      INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
     & MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
     & ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
     & FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
     & MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
     & DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
     & CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
     & CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
     & NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
     & PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
     & STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
     & ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
      COMMON/PARS/NUM,BUF,EOF
      INTEGER NUM,BUF(3)
      LOGICAL EOF
C
C          * REPORT INITIALIZATION PROCEDURES *
C
      CALL REPINT
      CALL NEWREP(.FALSE.)
C
C          * SPECIFY THE TITLE TO BE PRINTED AT THE TOP OF EACH PAGE
C            OF THE REPORT *
C
      CALL HEDING(27,27H** STATE AND CITY REPORT **,0,3)
C
C          * OPEN THE DATA BASE FILE AND DATA BASE TABLE FILE
C            ( LOGICAL I/O UNITS 2 AND 3 RESPECTIVELY *
C
      CALL OPEN(2,3,100,IERR)
C
C          * FIND STATE RECORD OCCURRENCE. IF NONE CAN BE FOUND,STOP *
C
 3    CALL FNM(SORDER,IERR)
      IF (IERR .EQ. -1) GOTO 2
C
C          * PLACE COMMENT INTO OUTPUT BUFFER *
C
      CALL BUFBLD(1,1,11,11HSTATE NAME-)
      * RETRIEVE STATE NAME, PLACE IN BUFFER, AND PRINT OUT *

CALL GFM(NAME,SORDER,BUF,IERR)
CALL BUFBLD(13,1,10,BUF)
CALL PBUF(2,0,.TRUE.)

      * PLACE COMMENT INTO OUTPUT BUFFER *

CALL BUFBLD(4,1,20,20HADMITTED IN THE YEAR)

      * RETRIEVE YEAR STATE WAS ADMITTED TO UNION, PLACE IN
        BUFFER, AND PRINT OUT *

CALL GFM(YEARAD,SORDER,NUM,IERR)
CALL NINBUF(NUM,26,4)
CALL PBUF(0,0,.TRUE.)

      * TEST TO SEE IF THERE ARE ANY CITY RECORDS RELATED TO
        THIS STATE. IF NOT, GO PROCESS NEXT STATE RECORD *

CALL SOM(CITIES,SORDER,IERR)
IF (IERR .EQ. -1) GOTO 3

      * PLACE COMMENT INTO OUTPUT BUFFER AND PRINT OUT *

CALL BUFBLD(4,1,32,32HMAJOR CITIES ARE      POPULATION)
CALL PBUF(0,0,.TRUE.)

      * RETRIEVE CITY NAME AND CITY POPULATION AND PUT INTO
        OUTPUT BUFFER *

CALL GFM(NAME,CITIES,BUF,IERR)
CALL BUFBLD(4,1,10,BUF)
CALL GFM(POP,CITIES,NUM,IERR)
CALL NINBUF(NUM,23,10)

      * PRINT OUT CITY NAME AND CITY POPULATION *

CALL PBUF(0,0,.TRUE.)

      * FIND NEXT CITY RECORD RELATED TO STATE. IF NONE, PROCESS
        NEXT STATE RECORD *

CALL FNM(CITIES,IERR)
IF (IERR .EQ. -1) GOTO 3
GOTO 1

      * CLOSE THE DATA BASE *

CALL CLOS(0,ARRAY,IERR)
END
```

TEST PROGRAM #3

This program gets an alphabetical listing of presidents. For each president it also finds out in what year and to whom the president was married by retrieving all occurrences of the marriage records associated to the president. It also finds all administration records associated to that president and for each administration record, the vice president related to that administration. If that isn't enough, the state the president was born in is also found and the state name is printed out.

```
C             **** PRESIDENTIAL INFORMATION REPORT PROGRAM ****
C
C             * THIS PROGRAM RETRIEVES INFORMATION ABOUT EACH  PRESIDENT
C               SUCH AS MARRIAGE DATE, WIFE'S NAME, YEARS OF ADMINISTRATION,
C               VICE PRESIDENTS AND STATE OF BIRTH *
C
C
C             * COMMON AREAS FOR THE PROGRAM *
C
       COMMON/NAMS/PRES,IDENT,FSTNAM,INITAL,SURNAM,
      & MONTHB,DAYB,YEARB,HEIGHT,PARTY,COLEGE,
      & ANSTRY,RELIGN,MONTHD,DAYD,YEARD,CAUSE,
      & FATHER,MOTHER,NOOCC,OCCUP,MARRGE,WIFE,
      & MONTHM,DAYM,YEARM,CHILDN,ADMIN,MONTH,
      & DAY,YEAR,VPRES,STATE,NAME,YEARAD,
      & CAPTAL,AREA,ARANK,POP,PRANK,VOTES,
      & CITY,ELECT,WINNER,OPPON,CONGRS,SENATE,
      & NUMBER,HSEREP,PRESME,PRESCS,PRESEN,
      & PRESAN,NUSTAT,CITIES,CABINT,SMEMBS,HMEMBS,
      & STASIZ,PORDER,SORDER,CORDER,EORDER,AORDER,
      & ELECTO,PRESS,NUB,CONGPS
       INTEGER PRES(2),IDENT(2),FSTNAM(2),INITAL(2),SURNAM(2),
      & MONTHB(2),DAYB(2),YEARB(2),HEIGHT(2),PARTY(2),COLEGE(2),
      & ANSTRY(2),RELIGN(2),MONTHD(2),DAYD(2),YEARD(2),CAUSE(2),
      & FATHER(2),MOTHER(2),NOOCC(2),OCCUP(2),MARRGE(2),WIFE(2),
      & MONTHM(2),DAYM(2),YEARM(2),CHILDN(2),ADMIN(2),MONTH(2),
      & DAY(2),YEAR(2),VPRES(2),STATE(2),NAME(2),YEARAD(2),
      & CAPTAL(2),AREA(2),ARANK(2),POP(2),PRANK(2),VOTES(2),
      & CITY(2),ELECT(2),WINNER(2),OPPON(2),CONGRS(2),SENATE(2),
      & NUMBER(2),HSEREP(2),PRESME(2),PRESCS(2),PRESEN(2),
      & PRESAN(2),NUSTAT(2),CITIES(2),CABINT(2),SMEMBS(2),HMEMBS(2),
      & STASIZ(2),PORDER(2),SORDER(2),CORDER(2),EORDER(2),AORDER(2),
      & ELECTO(2),PRESS(2),NUB(2),CONGPS(2)
       COMMON/PARS/NUM,BUF,EOF
       INTEGER NUM,BUF(3)
       LOGICAL EOF
C
C             * REPORT INITIALIZATION PROCEDURES *
C
       CALL REPINT
       CALL NEWREP(.FALSE.)
C
C             * SPECIFY TITLE TO BE PRINTED AT THE TOP OF EACH PAGE OF
C               THE REPORT *
C
       CALL HEDING(36,36H**PRESIDENTIAL INFORMATION REPORT **,0,3)
C
C             * OPEN THE DATA BASE FILE AND THE DATA BASE TABLE FILE
C               ( LOGICAL I/O UNITS 2 AND 3 RESPECTIVELY ) *
C
       CALL OPEN(2,3,100,IERR)
C
C             * FIND A PRESIDENT RECORD OCCURRENCE. IF NONE FOUND, STOP *
C
 1     CALL FNM(PORDER,IERR)
       IF (IERR .EQ. -1) GOTO 100
C
C             * PLACE COMMENT INTO OUTPUT BUFFER *
C
```

```
      IF (IERR .EQ. -1) GOTO 30

              * RETRIEVE VICE PRESIDENT'S LAST NAME AND STORE IN BUFFER *

      CALL GFM(SURNAM,CABINT,BUF,IERR)
      CALL BUFBLD(31,1,10,BUF)

              * PRINT OUT BUFFER WITH ADMINISTRATION DATA *

 0    CALL PBUF(0,0,.TRUE.)

              * TEST TO SEE IF THE PRESIDENT HAS ANY OTHER ADMINISTRATION
                RECORDS RELATED TO HIM. IF SO, PROCESS EACH RECORD AS
                ABOVE *

      CALL FNM(PRESAN,IERR)
      IF (IERR .GT. -1) GOTO 3

              * FIND STATE PRESIDENT WAS BORN IN, RETRIEVE NAME OF STATE
                AND STORE IN BUFFER WITH COMMENT *

      CALL SMM(PRESS,PORDER,IERR)
      CALL GFO(NAME,PRESS,BUF,IERR)
      CALL BUFBLD(1,1,8,8HBORN IN:)
      CALL BUFBLD(10,1,10,BUF)

              * PRINT OUT BUFFER WITH STATE INFORMATION *

      CALL PBUF(0,0,.TRUE.)

              * PROCESS NEXT PRESIDENT RECORD *

      GO TO 1

              * CLOSE THE DATA BASE *

100   CALL CLOS(0,ARRAY,IERR)
      END
      CALL BUFBLD(1,1,10,10HPRESIDENT-)
C
C            * RETRIEVE PRESIDENT'S LAST NAME, PUT IN OUTPUT BUFFER,
C              AND PRINT OUT BUFFER *
C
      CALL GFM(SURNAM,PORDER,BUF,IERR)
      CALL BUFBLD(12,1,10,BUF)
      CALL PBUF(2,0,.TRUE.)
C
C            * FIND A MARRIAGE RECORD RELATED TO THE PRESIDENT *
C
      CALL SOM(PRESME,PORDER,IERR)
C
C            * IF YEAR OF MARRIAGE EQUALS ZERO, A MESSAGE SHOULD BE
C              PRINTED THAT THE PRESIDENT WAS NOT MARRIED *
C
 2    CALL GFM(YEARM,PRESME,NUM,IERR)
      IF (NUM .EQ. 0) GOTO 5
C
C            * PUT COMMENT, YEAR OF MARRIAGE, AND WIFE'S NAME INTO
C              OUTPUT BUFFER AND PRINT OUT *
C
      CALL BUFBLD(1,1,22,22HWAS MARRIED IN      TO)
      CALL NINBUF(NUM,16,4)
      CALL GFM(WIFE,PRESME,BUF,IERR)
      CALL BUFBLD(24,1,10,BUF)
      CALL PBUF(0,0,.TRUE.)
C
C            * TEST TO SEE IF THERE ARE ANY MORE MARRIAGE RECORDS RELATED
C              TO THE PRESIDENT. IF SO, PROCESS EACH RECORD AS ABOVE *
C
      CALL FNM(PRESME,IERR)
      IF (IERR .GT. -1) GOTO 2
      GOTO 25
C
C            * PRINT MESSAGE THAT PRESIDENT WAS NOT MARRIED *
C
 5    CALL BUFBLD(1,1,15,15HWAS NOT MARRIED)
      CALL PBUF(0,0,.TRUE.)
C
C            * PUT COMMENTS INTO OUTPUT BUFFER AND PRINT OUT *
C
 25   CALL BUFBLD(1,1,25,25HHEADED ADMINISTRATIONS   )
      CALL BUFBLD(28,1,14,14HVICE PRESIDENT)
      CALL PBUF(0,0,.TRUE.)
C
C            * FIND AN ADMINISTRATION RECORD RELATED TO THE PRESIDENT *
C
      CALL SOM(PRESAN,PORDER,IERR)
C
C            * RETRIEVE YEAR OF ADMINISTRATION AND PUT IN BUFFER *
C
 3    CALL GFM(YEAR,PRESAN,NUM,IERR)
C
C            * TEST TO SEE IF THE PRESIDENT HAD A VICE PRESIDENT DURING
C              THE ADMINISTRATION. IF NOT, GO TO 30 *
C
      CALL BUFBLD(15,1,2,2HIN)
      CALL NINBUF(NUM,19,4)
      CALL SOM(CABINT,PRESAN,IERR)
```

APPENDIX A

Presidential File (PRESFILE) Listing

```
PRES      WASHINGT  WASHINGTONGEORGE              FEBRUARY          22      1732VIRGINIA  VIRGINIA
6FT. 2IN. FEDERALIST          ENGLISH   EPISCOPAL          3SURVEYOR  FARMER    SOLDIER   DECEMBER
        14      1799PNEUMONIA AUGUSTINE MARY               1MARTHA   JANUARY             6     1759
         0         2E1789     E1792              2A1        A2                 4C1       C2
C3        C4        PRES      ADAMSJ    ADAMS     JOHN                OCTOBER           30      1735
MASS.     MASS      5FT. 7IN. FEDERALISTHARVARD   ENGLISH   UNITARIAN          2LAWYER   TEACHER
JULY               4      1826DEBILITY  JOHN      SUSANNA            1ABIGAIL  OCTOBER          25
      1764         5         1E1796              1A3                 2C5       C6        PRES
JEFFERSO  JEFFERSON THOMAS              APRIL             13      1743VIRGINIA  VIRGINIA  6FT. 2IN.
DEM-REP   WM.-MARY  WELSH                        2LAWYER   WRITER    JULY                4     1826
DIARRHEA  PETER     JANE               1MARTHA   JANUARY             1      1772         6         2
E1800     E1804              2A4       A5                  4C7       C8        C9        C10
PRES      MADISON   MADISON   JAMES               MARCH             16      1751VIRGINIA  VIRGINIA
5FT. 4IN. DEM-REP   PRINCETON ENGLISH   EPISCOPAL          1LAWYER   JUNE              28      1836
DEBILITY  JAMES     ELEANOR            1DOLLEY   SEPTEMBER         15      1794         0         2
E1808     E1812              2A6       A7                  4C11      C12       C13       C14
PRES      MONROE    MONROE    JAMES               APRIL             28      1758VIRGINIA  VIRGINIA
6FT. 0IN. DEM-REP   WM.-MARY  SCOTCH    EPISCOPAL          2LAWYER   SOLDIER   JULY               4
      1831DEBILITY  SPENCE    ELIZABETH          1ELIZABETH FEBRUARY          16      1786         3
         2E1816    E1820               2A8       A9                  4C15      C16       C17
C18       PRES      ADAMSJQ   ADAMS     JOHN      Q.        JULY              11      1767MASS.
MASS      5FT. 7IN. DEM-REP   HARVARD   ENGLISH   UNITARIAN          2LAWYER   SECRETARY FEBRUARY
        23      1848PARALYSIS JOHN      ABIGAIL            1LOUISA   JULY              26      1797
         4         1E1824              1A10                2C19      C20       PRES      JACKSON
JACKSON   ANDREW              MARCH             15      1767S.C.      SC        6FT. 1IN. DEMOCRATIC
          SCOT-IRISHPRESBYT.           3LAWYER   SOLDIER   SADDLER   JUNE                8     1845
TB,DROPSY ANDREW    ELIZABETH          1RACHEL   AUGUST            15      1791         0         2
E1828     E1832              2A11      A12                 4C21      C22       C23       C24
PRES      VANBUREN  VAN BUREN MARTIN              DECEMBER            5      1782NEW YORK  NEWYORK
5FT. 6IN. DEMOCRATIC          DUTCH     DUTCH REF.         1LAWYER   JULY              24      1862
ASTHMA    ABRAHAM   MARIA              1HANNAH   FEBRUARY          21      1807         4         1
E1836              1A13                2C25      C26       PRES      HARRISOW  HARRISON  WILLIAM
H.        FEBRUARY           9      1773VIRGINIA  VIRGINIA  5FT. 8IN. WHIG      HAMP.-SYD.ENGLISH
EPISCOPAL          1SOLDIER  APRIL              4      1841PNEUMONIA BENJAMIN  ELIZABETH          1
ANNA      NOVEMBER          25      1795        10         1E1840              1A14               1
C27       PRES      TYLER     TYLER     JOHN                MARCH             29      1790VIRGINIA
VIRGINIA  6FT. 0IN. WHIG      WM.-MARY  ENGLISH   EPISCOPAL          1LAWYER   JANUARY           18
      1862FEVER     JOHN      MARY               2LETITIA  MARCH             29      1813         7
JULIA     JUNE              26      1844         7         0         1A15                2C27
C28       PRES      POLK      POLK      JAMES     K.        NOVEMBER           2      1795N.C.
NORTHC    5FT. 8IN. DEMOCRATICU.NC. CAR.SCOT-IRISHPRESBYT.           2LAWYER   CLERK     JUNE
        15      1849DIARRHEA  SAMUEL    JANE               1SARAH    JANUARY             1     1824
         0         1E1844              1A16                2C29      C30       PRES      TAYLOR
TAYLOR    ZACHARY             NOVEMBER          24      1784VIRGINIA  VIRGINIA  5FT. 8IN. WHIG
          ENGLISH   EPISCOPAL          2FARMER   SOLDIER
         1A17                1C31       PRES      FILLMORE  FILLMORE  MILLARD             JANUARY
         7      1800NEW YORK  NEWYORK   5FT. 9IN. WHIG                ENGLISH   UNITARIAN          2
LAWYER    WOOLCARDERMARCH              8      1874DEBILITY  NATHANIEL PHOEBE             2ABIGAIL
FEBRUARY           5      1826         2CAROLINE  FEBRUARY          10      1858         0         0
         1A18                2C31       C32       PRES      PIERCE    PIERCE    FRANKLIN
NOVEMBER          23      1804N.H.      NEWHAMP   5FT. 10IN.DEMOCRATICBOWDOIN   ENGLISH   EPISCOPAL
         1LAWYER    OCTOBER            8      1869STOM.INFL.BENJAMIN  ANNA                1JANE
NOVEMBER          10      1834         3          1E1852             1A19                2C33
C34       PRES      BUCHANAN  BUCHANAN  JAMES               APRIL             23      1791PA.
PA        6FT. 0IN. DEMOCRATICDICKINSON SCOT-IRISHPRESBYT.           1LAWYER    JUNE               1
      1868GOUT      JAMES     ELIZABETH          1                             0         0         0
         1E1856              1A20                2C35       C36       PRES      LINCOLN   LINCOLN
ABRAHAM             FEBRUARY          12      1809KENTUCKY  KENTUCKY  6FT. 4IN. REPUBLICAN
ENGLISH                      2LAWYER    FARM WORK.APRIL             15      1865ASSASSIN. THOMAS
NANCY              1MARY      NOVEMBER           4      1842         4         2E1860     E1864
         2A21       A22                3C37       C38       C39       PRES      JOHNSONA  JOHNSON
ANDREW              DECEMBER          29      1808N.C.      NORTHC    5FT. 10IN.REPUBLICAN
ENGLISH                      2TAILOR    PUB. OFF. JULY              31      1875PARALYSIS JACOB
MARY               1ELIZA     MAY                5      1827         5         0         1A23
         2C39       C40       PRES      GRANT     GRANT     ULYSSES   S.        APRIL             27
      1822OHIO      OHIO      5FT. 8IN. REPUBLICANU.S.MIL.ACENGL-SCOT METHODIST          1SOLDIER
JULY              23      1885CANCER    JESSE     HANNAH             1JULIA     AUGUST            22
      1848         4         2E1868     E1872              2A24       A25                4C41
C42       C43       C44       PRES      HAYES     HAYES     RUTHERFORDB.        OCTOBER            4
      1822OHIO      OHIO      5FT.8IN.  REPUBLICANKENYON              METHODIST          1LAWYER
JANUARY           17      1893HEART DIS.RUTHERFORDSOPHIA             1LUCY      DECEMBER          30
      1852         8      1E1876                 1A26                2C45       C46       PRES
GARFIELD  GARFIELD  JAMES     A.        NOVEMBER          19      1831OHIO      OHIO      6FT.0IN.
REPUBLICANWILLIAMS  ENGLISH   DIS.CHRIST         2LAWYER    CANAL DRIVSEPTEMBER         19      1881
ASSASSIN. ABRAM     ELIZA              1LUCRETIA  NOVEMBER          11      1858         7         1
E1880              1A27                1C47       PRES      ARTHUR    ARTHUR    CHESTER   A.
OCTOBER            5      1830VERMONT   VERMONT   6FT.2IN.  REPUBLICANUNION     SCOT-IRISHEPISCOPAL
         2LAWYER    TEACHER   NOVEMBER          18      1886APOPLEXY  WILLIAM   MALVINA            1
ELLEN     OCTOBER           25      1859         3         0         1A28                2C47
C48       PRES      CLEVELAN  CLEVELAND GROVER              MARCH             18      1837NEW JERSEY
NEWJERSEY 5FT.11IN. DEMOCRATIC          ENGL-IRISHPRESBYT.           2CLERK     TEACHER   JUNE
        24      1908DEBILITY  RICHARD   ANNA               1FRANCES   JUNE               2      1886
         5         2E1884     E1892              2A29       A31                4C49       C50
C53       C54       PRES      HARRISOB  HARRISON  BENJAMIN            AUGUST            20      1833
OHIO      OHIO      5FT. 6IN. REPUBLICANMIAMI O.  ENGLISH   PRESBYT.           1LAWYER    MARCH
        13      1901PNEUMONIA JOHN      ELIZABETH          2CAROLINE  OCTOBER           20      1853
        2MARY       APRIL              6      1896         1         1E1888              1A30
        2C51        C52       PRES      MCKINLEY  MCKINLEY  WILLIAM             JANUARY           29
      1843OHIO      OHIO      5FT.10IN. REPUBLICANALLEGHANY SCOT-IRISHMETHODIST          2LAWYER
TEACHER   SEPTEMBER         14      1901ASSASSIN. WILLIAM   NANCY              1IDA       JANUARY
        25      1871         2         2E1896     E1900              2A32       A33                3
```

```
C55       C56       C57       PRES      ROOSEVET  ROOSEVELT THEODORE            OCTOBER           27
      1858NEW YORK  NEWYORK   5FT. 10IN.REPUBLICANHARVARD   DUTCH     DUTCH REF.         2LAWYER
PUB.OFF.  JANUARY            6      1919RHEUMATISMTHEODORE  MARTHA             2ALICE     OCTOBER
        27      1880         1EDITH     DECEMBER           2      1886         5         1E1904
         2A34       A35                4C57       C58       C59       C60       PRES      TAFT
TAFT      WILLIAM   H.        SEPTEMBER         15      1857OHIO      OHIO      6FT. 0IN. REPUBLICAN
YALE      SCOT-IRISHUNITARIAN          1LAWYER    MARCH              8      1930DEBILITY  ALPHONSO
LOUISE             1HELEN     JUNE              19      1886         3         1E1908              1
A36                2C61       C62       PRES      WILSON    WILSON    WOODROW             DECEMBER
        28      1856VIRGINIA  VIRGINIA  6FT. 0IN. DEMOCRATICPRINCETON ENGLISH   PRESBYT.           2
LAWYER    TEACHER   FEBRUARY           3      1924HEART DIS.JOSEPH    JESSIE             2ELLEN
JUNE              24      1885         3EDITH     DECEMBER          18      1915         0         2
E1912     E1916              2A37       A38                4C63       C64       C65       C66
PRES      HARDING   HARDING   WARREN    G.        NOVEMBER           2      1865OHIO      OHIO
6FT. 0IN. REPUBLICANOHIO CENT.SCOT-IRISHBAPTIST            1JOURNALISTAUGUST             2      1923
EMBOLISM  GEORGE    PHOEBE             1FLORENCE  JULY              18      1891         0         1
E1920              1A39                2C67       C68       PRES      COOLIDGE  COOLIDGE  CALVIN
          JULY               4      1872VERMONT   VERMONT   5FT. 10IN.REPOBLICANAMHERST   ENGLISH
CONGREG.           1LAWYER    JANUARY            5      1933HEART ATT.JOHN      VICTORIA           1
GRACE     OCTOBER            4      1905         2         1E1924              2A40       A41
         3C68       C69       C70       PRES      HOOVER    HOOVER    HERBERT   C.        AUGUST
        10      1874IOWA      IOWA      5FT. 11IN.REPUBLICANSTANFORD  SWISS-GERMQUAKER             1
ENGINEER  OCTOBER           20      1964DEBILITY  JESSE     HULDA              1LOU       FEBRUARY
        10      1899         2         1E1928              1A42                2C71       C72
PRES      ROOSEVEF  ROOSEVELT FRANKLIN  D.        JANUARY           30      1882NEW YORK  NEWYORK
6FT. 2IN. DEMOCRATICHARVARD   DUTCH     EPISCOPAL          1LAWYER    APRIL             12      1945
CER.HEMORRJAMES     SARA               1ANNA      MARCH             17      1905         6         4
E1932     E1936     E1940     E1944              4A43       A44       A45       A46                7
C73       C74       C75       C76       C77       C78       C79       PRES      TRUMAN    TRUMAN
HARRY     S.        MAY                8      1884MISSOURI  MISSOURI  5FT. 9IN. DEMOCRATIC
ENGL-SCOT BAPTIST            2CLOTHIER  SENATOR                      0         0          JOHN
MARTHA             1ELIZABETH JUNE              28      1919         1         1E1948              2
A47       A48                4C79       C80       C81       C82       PRES      EISENHOW  EISENHOWER
DWIGHT    D.        OCTOBER           14      1890TEXAS     TEXAS     5FT. 10IN.REPUBLICANU.S.MIL.AC
SWISS-GERMPRESBYT.           1SOLDIER   MARCH             28      1969HEART DIS.DAVID     IDA
         1MAMIE     JULY               1      1916         2         2E1952     E1956              2
A49       A50                4C83       C84       C85       C86       PRES      KENNEDY   KENNEDY
JOHN      F.        MAY               29      1917MASS.     MASS      5FT. 11IN.DEMOCRATICHARVARD
IRISH     CATHOLIC           1SENATOR   NOVEMBER          22      1963ASSASSIN. JOSEPH    ROSE
         1JACQUELINESEPTEMBER         12      1953         3         1E1960              1A51
         2C87       C88       PRES      JCHNSONL  JOHNSON   LYNDON    B.        AUGUST            27
      1908TEXAS     TEXAS     6FT. 3IN. DEMOCRATICS.W.TEXAS ENGLISH   DIS.CHRIST         1SENATOR
                   0         0          SAM       REBEKAH            1CLAUDIA   NOVEMBER          17
      1934         2         1E1966
```

```
         3ADAMS     FEDERALIST        77CLINTON   DEM-REP           502 OTHERS                     5
ELECTION  E1796           1796J. ADAMS  FEDERALIST        71         4JEFFERSON DEM-REP           68
PINCKNEY  FEDERALIST        59BURR      DEM-REP           309 OTHERS                    48ELECTION
E1800           1800JEFFERSON DEM-REP           73         4BURR      DEM-REP            73ADAMS
FEDERALIST        65PINCKNEY  FEDERALIST        64JAY       FEDERALIST         1ELECTION  E1804
      1804JEFFERSON DEM-REP          162         1PINCKNEY  FEDERALIST        14ELECTION  E1808
      1808MADISON   DEM-REP          122         2PINCKNEY  FEDERALIST        47CLINTON   INDEP.
         6ELECTION  E1812           1812MADISON   DEM-REP          128         1CLINTON   INDEP.
        89ELECTION  E1816           1816MONROE    DEM-REP          183         1KING      FEDERALIST
        34ELECTION  E1820           1820MONROE    DEM-REP          231         1ADAMS     INDEP.
         1ELECTION  E1824           1824J.Q. ADAMSDEM-REP           84         3JACKSON   INDEP.
        99CRAWFORD  INDEP.            41CLAY      INDEP.           37ELECTION  E1828           1828
JACKSON   DEMOCRATIC       178         1ADAMS     NAT-REP          83ELECTION  E1832           1832
JACKSON   DEMOCRATIC       219         3CLAY      NAT-REP          49FLOYD     NULLIFIER         11
WIRT      ANTIMASON          7ELECTION  E1836           1836VAN BUREN DEMOCRATIC       170         4
HARRISON  WHIG              73WHITE     WHIG              26WEBSTER   WHIG              14MANGUM
INDEP.            11ELECTION  E1840           1840W.HARRISONWHIG             234         1VAN BUREN
DEMOCRATIC        60ELECTION  E1844           1844POLK      DEMOCRATIC       170         1CLAY
WHIG             105ELECTION  E1848           1848TAYLOR    WHIG             163         1CASS
DEMOCRATIC       127ELECTION  E1852           1852PIERCE    DEMOCRATIC       254         1SCOTT
WHIG              42ELECTION  E1856           1856BUCHANAN  DEMOCRATIC       174         2FREMONT
REPUBLICAN       114FILLMORE  AMERICAN           8ELECTION  E1860           1860LINCOLN   REPUBLICAN
       180         3DOUGLAS   DEMOCRATIC        12BRECKRIDGESOU. DEM.         72BELL      CONSTIT.
        39ELECTION  E1864           1864LINCOLN   REPUBLICAN       212         1MC CLELLANDEMOCRATIC
        21ELECTION  E1868           1868GRANT     REPUBLICAN       214         1SEYMOUR   DEMOCRATIC
        80ELECTION  E1872           1872GRANT     REPUBLICAN       286         1GREELEY   DEMOCRATIC
        66ELECTION  E1876           1876HAYES     REPUBLICAN       185         1TILDEN    DEMOCRATIC
       184ELECTION  E1880           1880GARFIELD  REPUBLICAN       214         1HANCOCK   DEMOCRATIC
       155ELECTION  E1884           1884CLEVELAND DEMOCRATIC       219         1BLAINE    REPUBLICAN
       182ELECTION  E1888           1888B.HARRISONREPUBLICAN       233         1CLEVELAND DEMOCRATIC
       168ELECTION  E1892           1892CLEVELAND DEMOCRATIC       277         2HARRISON  REPUBLICAN
       145WEAVER    PEOPLES           22ELECTION  E1896           1896MC KINLEY REPUBLICAN       271
         1BRYAN     DEMOCRATIC       176ELECTION  E1900           1900MC KINLEY REPUBLICAN       292
         1BRYAN     DEMOCRATIC       155ELECTION  E1904           1904TROOSEVELTREPUBLICAN       336
         1PARKER    DEMOCRATIC       140ELECTION  E1908           1908TAFT      REPUBLICAN       321
         1BRYAN     DEMOCRATIC       162ELECTION  E1912           1912WILSON    DEMOCRATIC       435
         2ROOSEVELT PROGRESS.         88TAFT      REPUBLICAN         8ELECTION  E1916           1916
WILSON    DEMOCRATIC       277         1HUGHES    REPUBLICAN       254ELECTION  E1920           1920
HARDING   REPUBLICAN       404         1COX       DEMOCRATIC       127ELECTION  E1924           1924
COOLIDGE  REPUBLICAN       382         2DAVIS     DEMOCRATIC       136LAFOLLETTEPROGRESS.          13
ELECTION  E1928           1928HOOVER    REPUBLICAN       444         1SMITH     DEMOCRATIC        87
ELECTION  E1932           1932FROOSEVELTDEMOCRATIC       472         1HOOVER    REPUBLICAN        59
ELECTION  E1936           1936FROOSEVELTDEMOCRATIC       523         1LANDON    REPUBLICAN         8
ELECTION  E1940           1940FROOSEVELTDEMOCRATIC       449         1WILLKE    REPUBLICAN        82
ELECTION  E1944           1944FROOSEVELTDEMOCRATIC       432         1DEWEY     REPUBLICAN        99
ELECTION  E1948           1948TRUMAN    DEMOCRATIC       303         2DEWEY     REPUBLICAN       189
```

```
THURMOND  STA.RIGHTS        39ELECTION  E1952           1952EISENHOWERREPUBLICAN       442
STEVENSON DEMOCRATIC        89ELECTION  E1956           1956EISENHOWERREPUBLICAN       457
STEVENSON DEMOCRATIC        73ELECTION  E1960           1960KENNEDY   DEMOCRATIC       303
NIXON     REPUBLICAN       219ELECTION  E1964           1964L. JOHNSONDEMOCRATIC       486
GOLDWATER REPUBLICAN        52ADMIN     A1        APRIL             30      1789WASHINGT
JOHN      ADAMS              2VERMONT   VERMONT   KENTUCKY  KENTUCKY  ADMIN     A2        MARCH
         4      1793WASHINGT           1JOHN      ADAMS              1TENNESSEE TENN      ADMIN
A3        MARCH              4      1797ADAMSJ             1THOMAS    JEFFERSON           0ADMIN
A4        MARCH              4      1801JEFFERSO           1AARON     BURR                1OHIO
OHIO      ADMIN     A5        MARCH              4      1805JEFFERSO           1GEORGE    CLINTON
         0ADMIN     A6        MARCH              4      1809MADISON            1GEORGE    CLINTON
         1LOUISIANA LA        ADMIN     A7        MARCH              4      1813MADISON            1
ELBRIDGE  GERRY              1INDIANA   INDIANA   ADMIN     A8        MARCH              4      1817
MONROE             1DANIEL    TOMPKINS           4MISS.     MISS      ILLINOIS  ILLINOIS  ALABAMA
ALABAMA   MAINE     MAINE     ADMIN     A9        MARCH              4      1821MONROE             1
DANIEL    TOMPKINS           1MISSOURI  MISSOURI  ADMIN     A10       MARCH              4      1825
ADAMSJQ            1JOHN      CALHOUN            0ADMIN     A11       MARCH              4      1829
JACKSON            1JOHN      CALHOUN            0ADMIN     A12       MARCH              4      1833
JACKSON            1MARTIN    VAN BUREN          2ARKANSAS  ARKANSAS  MICHIGAN  MICHIGAN  ADMIN
A13       MARCH              4      1837VANBUREN           1RICHARD   JOHNSON             0ADMIN
A14       MARCH              4      1841HARRISOW           1JOHN      TYLER               0ADMIN
A15       APRIL              6      1841TYLER              0         1FLORIDA   FLORIDA   ADMIN
A16       MARCH              4      1845PCLK               1GEORGE    DALLAS              3TEXAS
TEXAS     IOWA      IOWA      WISCONSIN WISC      ADMIN     A17       MARCH              4      1849
TAYLOR             1MILLARD   FILLMORE           0ADMIN     A18       JULY              10      1850
FILLMORE           0         1CALIFORNIACAL       ADMIN     A19       MARCH              4      1853
PIERCE             1WILLIAM   KING               0ADMIN     A20       MARCH              4      1857
BUCHANAN           1JOHN      BRECKRIDGE         3MINNESOTA MINN      OREGON    OREGON    KANSAS
KANSAS    ADMIN     A21       MARCH              4      1861LINCOLN            1HANNIBAL  HAMLIN
         2W.VA.     WESTVA    NEVADA    NEVADA    ADMIN     A22       MARCH              4      1865
LINCOLN            1ANDREW    JOHNSON            0ADMIN     A23       APRIL             15      1865
JOHNSONA           0         1NEBRASKA  NEBRASKA  ADMIN     A24       MARCH              4      1869
GRANT              1SCHUYLER  COLFAX             0ADMIN     A25       MARCH              4      1873
GRANT              1HENRY     WILSON             1COLORADO  COLORADO  ADMIN     A26       MARCH
         4      1877HAYES              1WILLIAM   WHEELER            0ADMIN     A27       MARCH
         4      1881GARFIELD           1CHESTER   ARTHUR             0ADMIN     A28       SEPTEMBER
        20      1881ARTHUR             0         0ADMIN     A29       MARCH              4      1885
CLEVELAN           1THOMAS    HENDRICKS          0ADMIN     A30       MARCH              4      1889
HARRISOB           1LEVI      MORTON             6WASHINGTONWASH      MONTANA   MONTANA   N.D.
NORTHD    S.D.      SD        IDAHO     IDAHO     WYOMING   WYOMING   ADMIN     A31       MARCH
         4      1893CLEVELAN           1ACLAI     STEVENSON          1UTAH      UTAH      ADMIN
A32       MARCH              4      1897MCKINLEY           1GARRET    HOBART              0ADMIN
A33       MARCH              4      1901MCKINLEY           1THEODORE  ROOSEVELT
A34       SEPTEMBER         14
```

```
NEWMEXICO ARIZONA   ARIZONA   ADMIN     A37       MARCH              4      1913WILSON             1
THOMAS    MARSHALL           OADMIN     A38       MARCH              4      1917WILSON             1
THOMAS    MARSHALL           OADMIN     A39       MARCH              4      1921HARDING            1
CALVIN    COOLIDGE           OADMIN     A40       AUGUST             3      1923COOLIDGE           0
         OADMIN     A41       MARCH            .4       1925COOLIDGE           1CHARLES   DAWES
         OADMIN     A42       MARCH             4       1929HOOVER             1CHARLES   CURTIS
         OADMIN     A43       MARCH             4       1933ROOSEVEF           1JOHN      GARNER
         OADMIN     A44       JANUARY          20       1937ROOSEVEF           1JOHN      GARNER
         OADMIN     A45       JANUARY          20       1941ROOSEVEF           1HENRY     WALLACE
         OADMIN     A46       JANUARY          20       1945ROOSEVEF           1HARRY     TRUMAN
         OADMIN     A47       APRIL            12       1945TRUMAN             0         OADMIN
A48       JANUARY           20      1949TRUMAN             1ALBEN     BARKLEY            OADMIN
A49       JANUARY           20      1953EISENHOW          1RICHARD   NIXON              OADMIN
A50       JANUARY           20      1957EISENHOW          1RICHARD   NIXON              2ALASKA
ALASKA    HAWAII    HAWAII    ADMIN     A51       JANUARY           20      1961KENNEDY            1
LYNDON    JOHNSON            OADMIN     A52       NOVEMBER          22      1963JOHNSONL           0
         OADMIN     A53       JANUARY          20       1965JOHNSONL           1HUBERT    HUMPHREY
         OCONGRESS  C1                 2FEDERALIST        17ANTI-FED.          9         2FEDERALIST
        38ANTI-FED.         26CONGRESS  C2                 2FEDERALIST        16DEM-REP          13
         2FEDERALIST        37DEM-REP           33CONGRESS  C3                 2FEDERALIST        17
DEM-REP           13         2FEDERALIST        48DEM-REP           57CONGRESS  C4                 2
FEDERALIST        19DEM-REP           13         2FEDERALIST        54DEM-REP           52CONGRESS
C5                 2FEDERALIST        20DEM-REP           12         2FEDERALIST        58DEM-REP
        48CONGRESS  C6                 2FEDERALIST        19DEM-REP           13         2FEDERALIST
        64DEM-REP           42CONGRESS  C7                 2FEDERALIST        14DEM-REP          18
         2FEDERALIST        36DEM-REP           69CONGRESS  C8                 2FEDERALIST         9
DEM-REP           25         2FEDERALIST        39DEM-REP          102CONGRESS  C9                 2
FEDERALIST         7DEM-REP           27         2FEDERALIST        25DEM-REP          116CONGRESS
C10                2FEDERALIST         6DEM-REP           28         2FEDERALIST        24DEM-REP
       118CONGRESS  C11                2FEDERALIST         6DEM-REP           28         2FEDERALIST
        48DEM-REP           94CONGRESS  C12                2FEDERALIST         6DEM-REP           30
         2FEDERALIST        36DEM-REP          108CONGRESS  C13                2FEDERALIST         9
DEM-REP           27         2FEDERALIST        68DEM-REP          112CONGRESS  C14                2
FEDERALIST        11DEM-REP           25         2FEDERALIST        65DEM-REP          117CONGRESS
C15                2FEDERALIST        10DEM-REP           34         2FEDERALIST        42DEM-REP
       141CONGRESS  C16                2FEDERALIST         7DEM-REP           35         2FEDERALIST
        27DEM-REP          156CONGRESS  C17                2FEDERALIST         4DEM-REP           44
         2FEDERALIST        25DEM-REP          158CONGRESS  C18                2FEDERALIST         4
DEM-REP           44         2FEDERALIST        26DEM-REP          187CONGRESS  C19                2
ADMIN.            26JACKSONIAN        20         2ADMIN.           105JACKSONIAN        97CONGRESS
C20                2ADMIN.            20JACKSONIAN        28         2ADMIN.            94JACKSONIAN
       119CONGRESS  C21                2DEMOCRATIC        26NAT-REP          22         2DEMOCRATIC
       139NAT-REP           74CONGRESS  C22                3DEMOCRATIC        25NAT-REP          21
OTHERS             2         3DEMOCRATIC       141NAT-REP           58OTHERS            14CONGRESS
C23                3DEMOCRATIC        20NAT-REP           20OTHERS             8         3DEMOCRATIC
       147ANTIMASON         53CTHERS            60CONGRESS  C24                2DEMOCRATIC        27
```

```
WHIG              25         2DEMOCRATIC       145WHIG              98CONGRESS  C25                3
DEMOCRATIC        30WHIG              18OTHERS             4         3DEMOCRATIC       108WHIG
       107OTHERS            24CONGRESS   C26                2DEMOCRATIC        28WHIG              22
         2DEMOCRATIC       124WHIG             118CONGRESS   C27                3DEMOCRATIC        22
WHIG              28OTHERS             2         3DEMOCRATIC       102WHIG             133OTHERS
         6CONGRESS   C28               3DEMOCRATIC        25WHIG              28OTHERS             1
         3DEMOCRATIC       142WHIG              79OTHERS             1CONGRESS  C29                2
DEMOCRATIC        31WHIG              25         3DEMOCRATIC       143WHIG              77OTHERS
         6CONGRESS   C30               3DEMOCRATIC        36WHIG              21OTHERS             1
         3DEMOCRATIC       108WHIG             115OTHERS              4CONGRESS  C31                3
DEMOCRATIC        35WHIG              25OTHERS             2         3DEMOCRATIC       112WHIG
       109OTHERS             9CONGRESS  C32                3DEMOCRATIC        38WHIG              24
OTHERS             3         3DEMOCRATIC       140WHIG              88OTHERS             5CONGRESS
C33                3DEMOCRATIC        40WHIG              22OTHERS             2         3DEMOCRATIC
       159WHIG              71OTHERS             4CONGRESS  C34                3DEMOCRATIC        36
REPUBLICAN        15OTHERS             5         3DEMOCRATIC        83REPUBLICAN        108OTHERS
        43CONGRESS   C35               3DEMOCRATIC        36REPUBLICAN         20OTHERS             8
         3DEMOCRATIC       118REPUBLICAN         92OTHERS            26CONGRESS   C36                3
DEMOCRATIC        10REPUBLICAN         26OTHERS             4         3DEMOCRATIC        92REPUBLICAN
       114OTHERS            31CONGRESS   C37                3DEMOCRATIC         9REPUBLICAN        31
OTHERS             8         3DEMOCRATIC        43REPUBLICAN        105OTHERS            30CONGRESS
C38                3DEMOCRATIC         9REPUBLICAN        36OTHERS             5         3DEMOCRATIC
        75REPUBLICAN        102OTHERS             9CONGRESS  C39                2DEMOCRATIC        10
REPUBLICAN        42         2DEMOCRATIC        42REPUBLICAN        149CONGRESS   C40                2
DEMOCRATIC        11REPUBLICAN         42         2DEMOCRATIC        49REPUBLICAN        143CONGRESS
C41                2DEMOCRATIC        11REPUBLICAN         56         2DEMOCRATIC        63REPUBLICAN
       149CONGRESS   C42               2DEMOCRATIC        17REPUBLICAN         52         3DEMOCRATIC
       104REPUBLICAN        134OTHERS             5CONGRESS  C43                3DEMOCRATIC        19
REPUBLICAN        49OTHERS             5         3DEMOCRATIC        92REPUBLICAN        194OTHERS
        14CONGRESS   C44               3DEMOCRATIC        29REPUBLICAN         45OTHERS             2
         3DEMOCRATIC       169REPUBLICAN        109OTHERS            14CONGRESS   C45                3
DEMOCRATIC        36REPUBLICAN         39OTHERS             1         2DEMOCRATIC       153REPUBLICAN
       140CONGRESS   C46               3DEMOCRATIC        42REPUBLICAN         33OTHERS             1
         3DEMOCRATIC       149REPUBLICAN        130OTHERS            14CONGRESS   C47                3
DEMOCRATIC        37REPUBLICAN         37OTHERS             1         3DEMOCRATIC       135REPUBLICAN
       147OTHERS            11CONGRESS   C48                3DEMOCRATIC        36REPUBLICAN        38
OTHERS             2         3DEMOCRATIC       197REPUBLICAN        118OTHERS            10CONGRESS
C49                2DEMOCRATIC        34REPUBLICAN         43         3DEMOCRATIC       183REPUBLICAN
       140OTHERS             2CONGRESS  C50                2DEMOCRATIC        37REPUBLICAN         39
         3DEMOCRATIC       169REPUBLICAN        152OTHERS             4CONGRESS  C51                2
DEMOCRATIC        44REPUBLICAN         38         2DEMOCRATIC       159REPUBLICAN        166CONGRESS
C52                3DEMOCRATIC        39REPUBLICAN        43OTHERS             2         3DEMOCRATIC
       235REPUBLICAN         88OTHERS             9CONGRESS  C53                3DEMOCRATIC        37
REPUBLICAN        39OTHERS
```

```
DEMOCRATIC        34REPUBLICAN        47CTHERS             7         3DEMOCRATIC       113REPUBLICAN
       204CTHERS           40CONGRESS  C56                3DEMOCRATIC        26REPUBLICAN        53
OTHERS             8         3DEMOCRATIC       163REPUBLICAN       185OTHERS             9CONGRESS
C57                3DEMOCRATIC        31REPUBLICAN        55OTHERS             4         3DEMOCRATIC
       151REPUBLICAN       197OTHERS             9CONGRESS  C58                2DEMOCRATIC        33
REPUBLICAN        57         2DEMOCRATIC       178REPUBLICAN       208CONGRESS  C59                2
DEMOCRATIC        33REPUBLICAN        57         2DEMOCRATIC       136REPUBLICAN       250CONGRESS
C60                2DEMOCRATIC        31REPUBLICAN        61         2DEMOCRATIC       164REPUBLICAN
       222CONGRESS  C61                2DEMOCRATIC        32REPUBLICAN        61         2DEMOCRATIC
       172REPUBLICAN       219CONGRESS  C62                2DEMOCRATIC        41REPUBLICAN        51
         3DEMOCRATIC       228REPUBLICAN       161OTHERS             1CONGRESS  C63                3
DEMOCRATIC        51REPUBLICAN        44OTHERS             1         3DEMOCRATIC       291REPUBLICAN
       127OTHERS            17CONGRESS  C64                2DEMOCRATIC        56REPUBLICAN        40
         3DEMOCRATIC       230REPUBLICAN       196OTHERS             9CONGRESS  C65                2
DEMOCRATIC        53REPUBLICAN        42         3DEMOCRATIC       216REPUBLICAN       210OTHERS
         6CONGRESS  C66                2DEMOCRATIC        47REPUBLICAN        49         3DEMOCRATIC
       190REPUBLICAN       240OTHERS             3CONGRESS  C67                2DEMOCRATIC        37
REPUBLICAN        59         3DEMOCRATIC       131REPUBLICAN       301OTHERS             1CONGRESS
C68                3DEMOCRATIC        43REPUBLICAN        51OTHERS             2         3DEMOCRATIC
       205REPUBLICAN       225CTHERS             5CONGRESS  C69                3DEMOCRATIC        39
REPUBLICAN        56OTHERS             1         3DEMOCRATIC       183REPUBLICAN       247OTHERS
         4CONGRESS  C70                3DEMOCRATIC        46REPUBLICAN        49OTHERS             1
         3DEMOCRATIC       195REPUBLICAN       237OTHERS             3CONGRESS  C71                3
DEMOCRATIC        39REPUBLICAN        56OTHERS             1         3DEMOCRATIC       167REPUBLICAN
       267OTHERS             1CONGRESS  C72                3DEMOCRATIC        47REPUBLICAN        48
OTHERS             1         3DEMOCRATIC       220REPUBLICAN       214OTHERS             1CONGRESS
C73                3DEMOCRATIC        60REPUBLICAN        35OTHERS             1         3DEMOCRATIC
       310REPUBLICAN       117CTHERS             5CONGRESS  C74                3DEMOCRATIC        69
REPUBLICAN        25OTHERS             2         3DEMOCRATIC       319REPUBLICAN       103OTHERS
        10CONGRESS  C75                3DEMOCRATIC        76REPUBLICAN        16OTHERS             4
         3DEMOCRATIC       331REPUBLICAN        89OTHERS            13CONGRESS  C76                3
DEMOCRATIC        69REPUBLICAN        23OTHERS             4         3DEMOCRATIC       261REPUBLICAN
       164OTHERS             4CONGRESS  C77                3DEMOCRATIC        66REPUBLICAN        28
OTHERS             2         3DEMOCRATIC       268REPUBLICAN       162OTHERS             5CONGRESS
C78                3DEMOCRATIC        58REPUBLICAN        37OTHERS             1         3DEMOCRATIC
       218REPUBLICAN       208OTHERS             4CONGRESS  C79                3DEMOCRATIC        56
REPUBLICAN        38OTHERS             1         3DEMOCRATIC       242REPUBLICAN       190OTHERS
         2CONGRESS  C80                2DEMOCRATIC        45REPUBLICAN        51         3DEMOCRATIC
       188REPUBLICAN       245CTHERS             1CONGRESS  C81                2DEMOCRATIC        54
REPUBLICAN        42         3DEMOCRATIC       263REPUBLICAN       171OTHERS             1CONGRESS
C82                2DEMOCRATIC        49REPUBLICAN        47         3DEMOCRATIC       234REPUBLICAN
       199OTHERS             1CCNGRESS  C83                3DEMOCRATIC        47REPUBLICAN        48
OTHERS             1         3DEMOCRATIC       211REPUBLICAN       221CTHERS             1CONGRESS
C84                3DEMOCRATIC        48REPUBLICAN        47OTHERS             1         2DEMOCRATIC
       232REPUBLICAN       203CONGRESS  C85                2DEMOCRATIC        49REPUBLICAN        47
         2DEMOCRATIC       233REPUBLICAN       200CONGRESS  C86                2DEMOCRATIC        64
```

```
REPUBLICAN        34         2DEMCCRATIC       283REPUBLICAN       153CONGRESS  C87               2
DEMOCRATIC        65REPUBLICAN        35         2DEMOCRATIC       263REPUBLICAN       174CONGRESS
C88                2DEMOCRATIC        67REPUBLICAN        33         2DEMOCRATIC       258REPUBLICAN
       177CONGRESS  C89                2DEMOCRATIC        68REPUBLICAN        32         2DEMOCRATIC
       295REPUBLICAN       140CONGRESS  C90                2DEMOCRATIC        64REPUBLICAN       36
         2DEMOCRATIC       248REPUBLICAN       187STATES    ALABAMA   ALABAMA         1819MONTGOMERY
     51609        29   3556000        21        10         4BIRMINGHAM    340887MOBILE        202779
MONTGOMERY    134393HUNTSVILLE    123519STATES    ALASKA    ALASKA          1959JUNEAU        586400
         1    277000        50         3         0STATES    ARIZONA   ARIZONA         1912PHOENIX
    113909         6   1670000        34         5         2PHOENIX       505666TUSCON        236877
STATES    ARKANSAS  ARKANSAS        1836LITTLEROCK     53104        28   2012000        31         6
         1LITTLEROCK    128929STATES    CAL       CALIFORNIA      1850SACRAMENTO    158693         3
  19221000         1        40        14LOSANGELES   2479015SAN FRAN.     740316SAN DIEGO     573224
OAKLAND       367548LONG BEACH    344168SACRAMENTO    237712SAN JOSE      204196FRESNO        133929
GLENDALE      119422PASADENA      116407BERKELEY      111268ANAHEIM       104184TORRANCE      100991
SANTA ANA     100350STATES    COLCRADC  CCLORADO        1876DENVER        104247         8  2048000
        30         6         1DENVER        493887STATES    CONN      CONN.           1788HARTFORD
      5009        48   2959000        24         8         4HARTFORD      162178BRIDGEPORT    156748
NEW HAVEN     141752WATERBURY     107130STATES    DELAWARE  DELAWARE        1787DOVER          2057
        49    534000        46         3         0STATES    FLORIDA   FLORIDA         1845TALLAHASSE
     58560        22   6160000         9        14         4MIAMI         291688TAMPA         274970
JACKSONV.     201030ST. PETER.    181298STATES    GEORGIA   GEORGIA         1788ATLANTA        58876
        21   4588000        15        12         3ATLANTA       487455SAVANNAH      149245COLUMBUS
    116779STATES    HAWAII    HAWAII          1959HONOLULU        6424        47   778000        40
         4         1HONOLULU      294194STATES    IDAHO     IDAHO           1890BOISE          83557
        13    705000        41         4         0STATES    ILLINOIS  ILLINOIS        1818SPRING.
     56400        25  10974000         4        26         3CHICAGO      3550404ROCKFORD      132109
PEORIA        103162STATES    INDIANA   INDIANA         1816INDIANAP.      36291        38  5067000
        12        13         6INDIANAP.     476258GARY          178320FORTWAYNE     172594EVANSVILLE
    144463SOUTH BEND    132445HAMMOND       111698STATES    IOWA      IOWA            1846DES MOINES
     56290        26   2748000        25         9         2DES MOINES    206739CED. RAP.     103545
STATES    KANSAS    KANSAS          1861TOPEKA         82264        14   2303000        29         7
         3WICHITA       254698KANSASCITY    121901TOPEKA        119484STATES    KENTUCKY  KENTUCKY
      1792FRANKFORT      40395        37   3229000        23         9         1LOUISVILLE    390639
STATES    LA        LOUISIANA       1812BATONROUGE     48523        31   3732000        19       10
         3NEWORLEANS    627525SHREVEPORT    160535BATONROUGE    154190STATES    MAINE     MAINE
      1820AUGUSTA        33215        39     979000       38         4         0STATES    MARYLAND
MARYLAND        1788ANNAPOLIS      10577        42   3757000        18        10         1BALTIMORE
    939024STATES    MASS      MASS.           1788BOSTON          8257        45   5437000       10
        14         5BOSTON        697197WORCESTER     186587SPRING.       174463CAMBRIDGE     107716
NEWBEDFORD    102447STATES    MICHIGAN  MICHIGAN        1837LANSING        58216        23  8740000
         7        21         5DETROIT      1670144FLINT         196940GR. RAPIDS    177313DEARBORN
    112007LANSING       107807STATES    MINN      MINNESOTA       1858ST. PAUL        84068       12
   3646000        20        10         3MINNEAP
   4627000        13        12         2ST. LOUIS     750026KANSASCITY    475539STATES    MONTANA
MONTANA         1889HELENA        147138         4    693000        43         4         OSTATES
NEBRASKA  NEBRASKA        1867LINCOLN        77227        15   1437000        35         5         2
OMAHA         301598LINCOLN       128521STATES   NEVADA     NEVADA          1864CARSONCITY    110540
         7    453000        47         3         OSTATES    NEWHAMP   N.H.            1788CONCORD
      9304        44    702000        42         4         OSTATES    NEWJERSEY NEW JERSEY      1787
TRENTON         7836        46   7078000         8        17         6NEWARK        405220JERSEYCITY
    276101PATERSON      143663CAMDEN        117159TRENTON       114167ELIZABETH     107698STATES
NEWMEXICO NEW MEXICO      1912SANTA FE      121666         5   1015000        37         4         1
ALBUQ.        201189STATES    NEWYORK   NEW YORK        1788ALBANY         49576        30  18113000
         2        43         8NEW YORK     7781984BUFFALO       532759ROCHESTER     318611SYRACUSE
    216038YONKERS       190634ALBANY        129726NIAG. FALL    102394UTICA         100410STATES
NORTHC    N.C.            1789RALEIGH        57712        24   5135000        11        13         4
CHARLOTTE     201564GREENSBORO    119574WIN. SALEM    111135RALEIGH       105722STATES    NORTHD
N.D.            1889BISMARK        70665        17    625000        45         4         OSTATES
OHIO      OHIO            1803COLUMBUS       41222        35  10591000         6        26         8
CLEVELAND     810858CINCINNATI    505550COLUMBUS      471316TOLEDO        318003AKRON         290351
DAYTON        262332YOUNGSTOWN    166689CANTON        113631STATES    OKLAHOMA  OKLAHOMA        1907
OKLA. CITY     69919        18   2518000        27         8         2OKLA. CITY    324253TULSA
    261685STATES    OREGON    OREGON          1859SALEM          96981        10   2008000        32
         6         1PORTLAND     372676STATES    PA        PA.             1787HARRISBURG     45333
        33  11712000         3        29         5PHILA.       2002512PITTSBURGH    604332ERIE
    138440SCRANTON      111443ALLENTOWN     108347STATES    RI        R.I.            1790PROVIDENCE
      1214        50    913000        39         4         1PROVIDENCE    207498STATES    SC
S.C.            1788COLUMBIA       31055        40   2692000        26         8         OSTATES
SD        S.D.            1889PIERRE         77047        16    657000        44         4         0
STATES    TENN      TENNESSEE      1796NASHVILLE      42244        34   3976000        17        11
         4MEMPHIS       536585NASHVILLE     170874CHATANOOGA    130009KNOXVILLE     118827STATES
TEXAS     TEXAS           1845AUSTIN        267339         2  10972000         5        25        11
HOUSTON       939219DALLAS        679684SANANTONIO    587718FORTWORTH     356268ELPASO        276687
AUSTIN        185545CORP. CHR.    167690AMARILLO      137969LUBBOCK       128691BEAUMONT      119175
MICH. FALL    101724STATES    UTAH      UTAH            1896S.L. CITY      84916        11   1034000
        36         4         1S.L. CITY     189454STATES    VERMONT   VERMONT         1791MONTPELIER
      9609        43    422000        48         3         OSTATES    VIRGINIA  VIRGINIA        1788
RICHMOND       40815        36   4597000        14        12         4NORFOLK       304869RICHMOND
    219958PORTSMOUTH    144773NEWP. NEWS    113662STATES    WASH      WASHINGTON      1889OLYMPIA
     68192        20   3276000        22         9         3SEATTLE       557087SPOKANE       181608
TACOMA        147979STATES    WESTVA    W.VA            1863CHARLESTON     24181        41   1805000
        33         7         OSTATES    WISC      WISCONSIN      1848MADISON        56154        27
   4213000        16        12         2MILWAUKEE     741324MADISON       157844STATES    WYOMING
WYOMING         1890CHEYENNE       97914         9    315000        49         3          0*********
```

APPENDIX B

Presidential Report Examples

```
          PSA VERSION 740/10                                        SEP 29, 1974  17:17.10      PAGE    1
                                          UNIVERSITY OF MICHIGAN / MTS

                                         ** PRESIDENT/ELECTION REPORT **


      PRESIDENT NAME JOHN           ADAMS
         WON ELECTION OF  1796 AGAINST JEFFERSON


      PRESIDENT NAME JOHN        Q. ADAMS
         WON ELECTION OF  1824 AGAINST JACKSON


      PRESIDENT NAME CHESTER     A. ARTHUR


      PRESIDENT NAME JAMES          BUCHANAN
         WON ELECTION OF  1856 AGAINST FREMONT


      PRESIDENT NAME GROVER         CLEVELAND
         WON ELECTION OF  1884 AGAINST BLAINE
         WON ELECTION OF  1892 AGAINST HARRISON


      PRESIDENT NAME CALVIN         COOLIDGE
         WON ELECTION OF  1924 AGAINST DAVIS


      PRESIDENT NAME DWIGHT      D. EISENHOWER
         WON ELECTION OF  1952 AGAINST STEVENSON
         WON ELECTION OF  1956 AGAINST STEVENSON


      PRESIDENT NAME MILLARD        FILLMORE


      PRESIDENT NAME JAMES       A. GARFIELD
         WON ELECTION OF  1880 AGAINST HANCOCK


      PRESIDENT NAME ULYSSES     S. GRANT
         WON ELECTION OF  1868 AGAINST SEYMOUR
```

```
     PSA VERSION 740.10                UNIVERSITY OF MICHIGAN / MTS

                                     ** PRESIDENT/ELECTION REPORT **


PRESIDENT NAME WARREN        G. HARDING
   WON ELECTION OF  1920 AGAINST COX


PRESIDENT NAME BENJAMIN         HARRISON
   WON ELECTION OF  1888 AGAINST CLEVELAND


PRESIDENT NAME WILLIAM       H. HARRISON
   WON ELECTION OF  1840 AGAINST VAN BUREN


PRESIDENT NAME RUTHERFORD  B. HAYES
   WON ELECTION OF  1876 AGAINST TILDEN


PRESIDENT NAME HERBERT       C. HOOVER
   WON ELECTION OF  1928 AGAINST SMITH


PRESIDENT NAME ANDREW           JACKSON
   WON ELECTION OF  1828 AGAINST ADAMS
   WON ELECTION OF  1832 AGAINST CLAY


PRESIDENT NAME THOMAS           JEFFERSON
   WON ELECTION OF  1800 AGAINST BURR
   WON ELECTION OF  1804 AGAINST PINCKNEY


PRESIDENT NAME ANDREW           JOHNSON


PRESIDENT NAME LYNDON        B. JOHNSON
   WON ELECTION OF  1964 AGAINST GOLDWATER


PRESIDENT NAME JOHN          F. KENNEDY
   WON ELECTION OF  1960 AGAINST NIXON
```

```
     PSA VERSION 740710                                           SEP 29, 1974  17:17.10      PAGE     3
                                      UNIVERSITY OF MICHIGAN / MTS

                                    ** PRESIDENT/ELECTION REPORT **


PRESIDENT NAME ABRAHAM          LINCOLN
   WON ELECTION OF  1860 AGAINST DOUGLAS
   WON ELECTION OF  1864 AGAINST MC CLELLAN


PRESIDENT NAME JAMES            MADISON
   WON ELECTION OF  1808 AGAINST PINCKNEY
   WON ELECTION OF  1812 AGAINST CLINTON


PRESIDENT NAME WILLIAM          MCKINLEY
   WON ELECTION OF  1896 AGAINST BRYAN
   WON ELECTION OF  1900 AGAINST BRYAN


PRESIDENT NAME JAMES            MONROE
   WON ELECTION OF  1816 AGAINST KING
   WON ELECTION OF  1820 AGAINST ADAMS


PRESIDENT NAME FRANKLIN         PIERCE
   WON ELECTION OF  1852 AGAINST SCOTT


PRESIDENT NAME JAMES         K. POLK
   WON ELECTION OF  1844 AGAINST CLAY


PRESIDENT NAME FRANKLIN      D. ROOSEVELT
   WON ELECTION OF  1932 AGAINST HOOVER
   WON ELECTION OF  1936 AGAINST LANDON
   WON ELECTION OF  1940 AGAINST WILLKE
   WON ELECTION OF  1944 AGAINST DEWEY


PRESIDENT NAME THEODORE         ROOSEVELT
   WON ELECTION OF  1904 AGAINST PARKER
```

```
                    UNIVERSITY OF MICHIGAN / MTS

                  ** PRESIDENT/ELECTION REPORT **


PRESIDENT NAME WILLIAM     H. TAFT
   WON ELECTION OF  1908 AGAINST BRYAN

PRESIDENT NAME ZACHARY          TAYLOR
   WON ELECTION OF  1848 AGAINST CASS

PRESIDENT NAME HARRY       S. TRUMAN
   WON ELECTION OF  1948 AGAINST DEWEY

PRESIDENT NAME JOHN             TYLER

PRESIDENT NAME MARTIN           VAN BUREN
   WON ELECTION OF  1836 AGAINST HARRISON

PRESIDENT NAME GEORGE           WASHINGTON
   WON ELECTION OF  1789 AGAINST ADAMS
   WON ELECTION OF  1792 AGAINST ADAMS

PRESIDENT NAME WOODROW          WILSON
   WON ELECTION OF  1912 AGAINST ROOSEVELT
   WON ELECTION OF  1916 AGAINST HUGHES
```

```
      PSA VERSION 740710                                   OCT  6, 1974  16:48.02      PAGE      1
                                  UNIVERSITY OF MICHIGAN / MTS

                                    ** STATE AND CITY REPORT **


STATE NAME- ALABAMA
   ADMITTED IN THE YEAR  1819
   MAJOR CITIES ARE      POPULATION
   BIRMINGHAM             340887
   MOBILE                 202779
   MONTGOMERY             134393
   HUNTSVILLE             123519


STATE NAME- ALASKA
   ADMITTED IN THE YEAR  1959


STATE NAME- ARIZONA
   ADMITTED IN THE YEAR  1912
   MAJOR CITIES ARE      POPULATION
   PHOENIX                505666
   TUSCON                 236877


STATE NAME- ARKANSAS
   ADMITTED IN THE YEAR  1836
   MAJOR CITIES ARE      POPULATION
   LITTLEROCK             128929


STATE NAME- CALIFORNIA
   ADMITTED IN THE YEAR  1850
   MAJOR CITIES ARE      POPULATION
   LOSANGELES            2479015
   SAN FRAN.              740316
   SAN DIEGO              573224
   OAKLAND                367548
   LONG BEACH             344168
   SACRAMENTO             237712
   SAN JOSE               204196
   FRESNO                 133929
   GLENDALE               119422
```

```
                         UNIVERSITY OF MICHIGAN / MTS

                         ** STATE AND CITY REPORT **


   BERKELEY              111268
   ANAHEIM               104184
   TORRANCE              100991
   SANTA ANA             100350


STATE NAME- COLORADO
   ADMITTED IN THE YEAR  1876
   MAJOR CITIES ARE      POPULATION
   DENVER                 493887


STATE NAME- CONN.
   ADMITTED IN THE YEAR  1788
   MAJOR CITIES ARE      POPULATION
   HARTFORD               162178
   BRIDGEPORT             156748
   NEW HAVEN              141752
   WATERBURY              107130


STATE NAME- DELAWARE
   ADMITTED IN THE YEAR  1787


STATE NAME- FLORIDA
   ADMITTED IN THE YEAR  1845
   MAJOR CITIES ARE      POPULATION
   MIAMI                  291688
   TAMPA                  274970
   JACKSONV.              201030
   ST. PETER.             181298


STATE NAME- GEORGIA
   ADMITTED IN THE YEAR  1788
   MAJOR CITIES ARE      POPULATION
   ATLANTA                487455
   SAVANNAH               149245
```

```
                           ** STATE AND CITY REPORT **


   COLUMBUS                116779


STATE NAME- HAWAII
   ADMITTED IN THE YEAR  1959
   MAJOR CITIES ARE      POPULATION
   HONOLULU                294194


STATE NAME- IDAHO
   ADMITTED IN THE YEAR  1890


STATE NAME- ILLINOIS
   ADMITTED IN THE YEAR  1818
   MAJOR CITIES ARE      POPULATION
   CHICAGO               3550404
   ROCKFORD                132109
   PEORIA                  103162


STATE NAME- INDIANA
   ADMITTED IN THE YEAR  1816
   MAJOR CITIES ARE      POPULATION
   INDIANAP.               476258
   GARY                    178320
   FORTWAYNE               172594
   EVANSVILLE              144463
   SOUTH BEND              132445
   HAMMOND                 111698


STATE NAME- IOWA
   ADMITTED IN THE YEAR  1846
   MAJOR CITIES ARE      POPULATION
   DES MOINES              206739
   CED. RAP.               103545
```

```
                                 ** STATE AND CITY REPORT **

STATE NAME- KANSAS
   ADMITTED IN THE YEAR  1861
   MAJOR CITIES ARE      POPULATION
   WICHITA                254698
   KANSASCITY             121901
   TOPEKA                 119484

STATE NAME- KENTUCKY
   ADMITTED IN THE YEAR  1792
   MAJOR CITIES ARE      POPULATION
   LOUISVILLE             390639

STATE NAME- LOUISIANA
   ADMITTED IN THE YEAR  1812
   MAJOR CITIES ARE      POPULATION
   NEWORLEANS             627525
   SHREVEPORT             160535
   BATONROUGE             154190

STATE NAME- MAINE
   ADMITTED IN THE YEAR  1820

STATE NAME- MARYLAND
   ADMITTED IN THE YEAR  1788
   MAJOR CITIES ARE      POPULATION
   BALTIMORE              939024

STATE NAME- MASS.
   ADMITTED IN THE YEAR  1788
   MAJOR CITIES ARE      POPULATION
   BOSTON                 697197
   WORCESTER              186587
   SPRING.                174463
   CAMBRIDGE              107716
```

```
       PSA VERSION 740710                                              OCT  6, 1974  16:48.02       PAGE       5
                                    UNIVERSITY OF MICHIGAN / MTS

                                    ** STATE AND CITY REPORT **


   NEWBEDFORD             102447


STATE NAME- MICHIGAN
   ADMITTED IN THE YEAR  1837
   MAJOR CITIES ARE      POPULATION
   DETROIT               1670144
   FLINT                  196940
   GR. RAPIDS             177313
   DEARBORN               112007
   LANSING                107807


STATE NAME- MINNESOTA
   ADMITTED IN THE YEAR  1858
   MAJOR CITIES ARE      POPULATION
   MINNEAP.               482872
   ST. PAUL               313411
   DULUTH                 106884


STATE NAME- MISS.
   ADMITTED IN THE YEAR  1817
   MAJOR CITIES ARE      POPULATION
   JACKSON                144422


STATE NAME- MISSOURI
   ADMITTED IN THE YEAR  1812
   MAJOR CITIES ARE      POPULATION
   ST. LOUIS              750026
   KANSASCITY             475539


STATE NAME- MONTANA
   ADMITTED IN THE YEAR  1889
```

```
                    ** STATE AND CITY REPORT **


   ADMITTED IN THE YEAR  1867
   MAJOR CITIES ARE      POPULATION
   OMAHA                  301598
   LINCOLN                128521


STATE NAME- NEVADA
   ADMITTED IN THE YEAR  1864


STATE NAME- N.H.
   ADMITTED IN THE YEAR  1788


STATE NAME- NEW JERSEY
   ADMITTED IN THE YEAR  1787
   MAJOR CITIES ARE      POPULATION
   NEWARK                 405220
   JERSEYCITY             276101
   PATERSON               143663
   CAMDEN                 117159
   TRENTON                114167
   ELIZABETH              107698


STATE NAME- NEW MEXICO
   ADMITTED IN THE YEAR  1912
   MAJOR CITIES ARE      POPULATION
   ALBUQ.                 201189


STATE NAME- NEW YORK
   ADMITTED IN THE YEAR  1788
   MAJOR CITIES ARE      POPULATION
   NEW YORK              7781984
   BUFFALO                532759
   ROCHESTER              318611
   SYRACUSE               216038
   YONKERS                190634
   PSA VERSION 740710                                            OCT  6, 1974  16:48.02       PAGE
                                  UNIVERSITY OF MICHIGAN / MTS

                                   ** STATE AND CITY REPORT **


    ALBANY                  129726
    NIAG. FALL              102394
    UTICA                   100410


STATE NAME- N.C.
    ADMITTED IN THE YEAR  1789
    MAJOR CITIES ARE      POPULATION
    CHARLOTTE               201564
    GREENSBORO              119574
    WIN. SALEM              111135
    RALEIGH                 105722


STATE NAME- N.D.
    ADMITTED IN THE YEAR  1889


STATE NAME- OHIO
    ADMITTED IN THE YEAR  1803
    MAJOR CITIES ARE      POPULATION
    CLEVELAND               810858
    CINCINNATI              505550
    COLUMBUS                471316
    TOLEDO                  318003
    AKRON                   290351
    DAYTON                  262332
    YOUNGSTOWN              166689
    CANTON                  113631


STATE NAME- OKLAHOMA
    ADMITTED IN THE YEAR  1907
    MAJOR CITIES ARE      POPULATION
    OKLA. CITY              324253
    TULSA                   261685


STATE NAME- OREGON
```

```
                         UNIVERSITY OF MICHIGAN / MTS

                         ** STATE AND CITY REPORT **


   ADMITTED IN THE YEAR  1859
   MAJOR CITIES ARE      POPULATION
   PORTLAND               372676


STATE NAME- PA.
   ADMITTED IN THE YEAR  1787
   MAJOR CITIES ARE      POPULATION
   PHILA.                2002512
   PITTSBURGH             604332
   ERIE                   138440
   SCRANTON               111443
   ALLENTOWN              108347


STATE NAME- R.I.
   ADMITTED IN THE YEAR  1790
   MAJOR CITIES ARE      POPULATION
   PROVIDENCE             207498


STATE NAME- S.C.
   ADMITTED IN THE YEAR  1788


STATE NAME- S.D.
   ADMITTED IN THE YEAR  1889


STATE NAME- TENNESSEE
   ADMITTED IN THE YEAR  1796
   MAJOR CITIES ARE      POPULATION
   MEMPHIS                536585
   NASHVILLE              170874
   CHATANOOGA             130009
   KNOXVILLE              118827


STATE NAME- TEXAS
```

```
PSA VERSION 740710                                        OCT  6, 1974  16:48.02       PAGE       9
                                  UNIVERSITY OF MICHIGAN / MTS

                                     ** STATE AND CITY REPORT **


   ADMITTED IN THE YEAR  1845
   MAJOR CITIES ARE      POPULATION
   HOUSTON                939219
   DALLAS                 679684
   SANANTONIO             587718
   FORTWORTH              356268
   ELPASO                 276687
   AUSTIN                 185545
   CORP. CHR.             167690
   AMARILLO               137969
   LUBBOCK                128691
   BEAUMONT               119175
   MICH. FALL             101724


STATE NAME- UTAH
   ADMITTED IN THE YEAR  1896
   MAJOR CITIES ARE      POPULATION
   S.L. CITY              189454


STATE NAME- VERMONT
   ADMITTED IN THE YEAR  1791


STATE NAME- VIRGINIA
   ADMITTED IN THE YEAR  1788
   MAJOR CITIES ARE      POPULATION
   NORFOLK                304869
   RICHMOND               219958
   PORTSMOUTH             144773
   NEWP. NEWS             113662


STATE NAME- WASHINGTON
   ADMITTED IN THE YEAR  1889
   MAJOR CITIES ARE      POPULATION
   SEATTLE                557087
   SPOKANE
```

```
                         UNIVERSITY OF MICHIGAN / MTS

                         ** STATE AND CITY REPORT **


   TACOMA                      147979


STATE NAME- W.VA
   ADMITTED IN THE YEAR  1863


STATE NAME- WISCONSIN
   ADMITTED IN THE YEAR  1848
   MAJOR CITIES ARE      POPULATION
   MILWAUKEE              741324
   MADISON                157844


STATE NAME- WYOMING
   ADMITTED IN THE YEAR  1890
```

```
        PSA VERSION 740710                                 OCT   6, 1974   19:52.18      PAGE      1
                                    UNIVERSITY OF MICHIGAN / MTS

                                 **PRESIDENTIAL INFORMATION REPORT **


PRESIDENT- ADAMS
WAS MARRIED IN 1764 TO ABIGAIL
HEADED ADMINISTRATIONS     VICE PRESIDENT
           IN  1797           JEFFERSON
BORN IN: MASS.


PRESIDENT- ADAMS
WAS MARRIED IN 1797 TO LOUISA
HEADED ADMINISTRATIONS     VICE PRESIDENT
           IN  1825           CALHOUN
BORN IN: MASS.


PRESIDENT- ARTHUR
WAS MARRIED IN 1859 TO ELLEN
HEADED ADMINISTRATIONS     VICE PRESIDENT
           IN  1881
BORN IN: VERMONT


PRESIDENT- BUCHANAN
WAS NOT MARRIED
HEADED ADMINISTRATIONS     VICE PRESIDENT
           IN  1857           BRECKRIDGE
BORN IN: PA.


PRESIDENT- CLEVELAND
WAS MARRIED IN 1886 TO FRANCES
HEADED ADMINISTRATIONS     VICE PRESIDENT
           IN  1885           HENDRICKS
           IN  1893           STEVENSON
BORN IN: NEW JERSEY


PRESIDENT- COOLIDGE
WAS MARRIED IN 1905 TO GRACE
```

```
                        **PRESIDENTIAL INFORMATION REPORT **


              IN  1923
              IN  1925        DAWES
BORN IN: VERMONT


PRESIDENT- EISENHOWER
WAS MARRIED IN 1916 TO MAMIE
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1953        NIXON
              IN  1957        NIXON
BORN IN: TEXAS


PRESIDENT- FILLMORE
WAS MARRIED IN 1826 TO ABIGAIL
WAS MARRIED IN 1858 TO CAROLINE
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1850
BORN IN: NEW YORK


PRESIDENT- GARFIELD
WAS MARRIED IN 1858 TO LUCRETIA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1881        ARTHUR
BORN IN: OHIO


PRESIDENT- GRANT
WAS MARRIED IN 1848 TO JULIA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1869        COLFAX
              IN  1873        WILSON
BORN IN: OHIO


PRESIDENT- HARDING
WAS MARRIED IN 1891 TO FLORENCE
HEADED ADMINISTRATIONS     VICE PRESIDENT
```

```
        PSA VERSION 740710                              OCT  6, 1974  19:52.18      PAGE     3
                                     UNIVERSITY OF MICHIGAN / MTS

                                    **PRESIDENTIAL INFORMATION REPORT **


              IN  1921         COOLIDGE
BORN IN: OHIO


PRESIDENT- HARRISON
WAS MARRIED IN 1853 TO CAROLINE
WAS MARRIED IN 1896 TO MARY
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1889         MORTON
BORN IN: OHIO


PRESIDENT- HARRISON
WAS MARRIED IN 1795 TO ANNA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1841         TYLER
BORN IN: VIRGINIA


PRESIDENT- HAYES
WAS MARRIED IN 1852 TO LUCY
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1877         WHEELER
BORN IN: OHIO


PRESIDENT- HOOVER
WAS MARRIED IN 1899 TO LOU
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1929         CURTIS
BORN IN: IOWA


PRESIDENT- JACKSON
WAS MARRIED IN 1791 TO RACHEL
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1829         CALHOUN
              IN  1833         VAN BUREN
BORN IN: S C
```

```
                    **PRESIDENTIAL INFORMATION REPORT **


PRESIDENT- JEFFERSON
WAS MARRIED IN 1772 TO MARTHA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1801        BURR
              IN  1805        CLINTON
BORN IN: VIRGINIA


PRESIDENT- JOHNSON
WAS MARRIED IN 1827 TO ELIZA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1865
BORN IN: N.C.


PRESIDENT- JOHNSON
WAS MARRIED IN 1934 TO CLAUDIA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1963
              IN  1965        HUMPHREY
BORN IN: TEXAS


PRESIDENT- KENNEDY
WAS MARRIED IN 1953 TO JACQUELINE
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1961        JOHNSON
BORN IN: MASS.


PRESIDENT- LINCOLN
WAS MARRIED IN 1842 TO MARY
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1861        HAMLIN
              IN  1865        JOHNSON
BORN IN: KENTUCKY


PRESIDENT- MADISON
```

```
            **PRESIDENTIAL INFORMATION REPORT **


WAS MARRIED IN 1794 TO DOLLEY
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1809        CLINTON
              IN  1813        GERRY
BORN IN: VIRGINIA


PRESIDENT- MCKINLEY
WAS MARRIED IN 1871 TO IDA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1897        HOBART
              IN  1901        ROOSEVELT
BORN IN: OHIO


PRESIDENT- MONROE
WAS MARRIED IN 1786 TO ELIZABETH
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1817        TOMPKINS
              IN  1821        TOMPKINS
BORN IN: VIRGINIA


PRESIDENT- PIERCE
WAS MARRIED IN 1834 TO JANE
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1853        KING
BORN IN: N.H.


PRESIDENT- POLK
WAS MARRIED IN 1824 TO SARAH
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1845        DALLAS
BORN IN: N.C.


PRESIDENT- ROOSEVELT
WAS MARRIED IN
```

```
                    **PRESIDENTIAL INFORMATION REPORT **


HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1933        GARNER
              IN  1937        GARNER
              IN  1941        WALLACE
              IN  1945        TRUMAN
BORN IN: NEW YORK


PRESIDENT- ROOSEVELT
WAS MARRIED IN 1880 TO ALICE
WAS MARRIED IN 1886 TO EDITH
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1901
              IN  1905        FAIRBANKS
BORN IN: NEW YORK


PRESIDENT- TAFT
WAS MARRIED IN 1886 TO HELEN
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1909        SHERMAN
BORN IN: OHIO


PRESIDENT- TAYLOR
WAS MARRIED IN 1810 TO MARGARET
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1849        FILLMORE
BORN IN: VIRGINIA


PRESIDENT- TRUMAN
WAS MARRIED IN 1919 TO ELIZABETH
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1945
              IN  1949        BARKLEY
BORN IN: MISSOURI
```

```
                                        **PRESIDENTIAL INFORMATION REPORT **


PRESIDENT- TYLER
WAS MARRIED IN 1813 TO LETITIA
WAS MARRIED IN 1844 TO JULIA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1841
BORN IN: VIRGINIA


PRESIDENT- VAN BUREN
WAS MARRIED IN 1807 TO HANNAH
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1837        JOHNSON
BORN IN: NEW YORK


PRESIDENT- WASHINGTON
WAS MARRIED IN 1759 TO MARTHA
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1789        ADAMS
              IN  1793        ADAMS
BORN IN: VIRGINIA


PRESIDENT- WILSON
WAS MARRIED IN 1885 TO ELLEN
WAS MARRIED IN 1915 TO EDITH
HEADED ADMINISTRATIONS     VICE PRESIDENT
              IN  1913        MARSHALL
              IN  1917        MARSHALL
BORN IN: VIRGINIA
```