MIT/LCS/TM-60

# CAMAC:
# GROUP MANIPULATION SYSTEM

Randell B. Weiss

March 1975

TM-60

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Project MAC

# CAMAC: GROUP MANIPULATION SYSTEM

Randell B. Weiss

ABSTRACT: CAMAC is a collection of group manipulation programs with an easy to use interface. With groups defined by either generating permutations or generators and relations the system can find coset tables, normalizers, centralizers, stabilizers, orbits, conjugacy classes, and isomorph classes of combinatorial objects, etc.

This research was supported by the National Science Foundation under research grant no. P4P0460-000.

March 1975

How to use the CAMAC system:

Background:

This section is addressed to the user who is unfamiliar with computers. Even if you are knowledgeable in this area, we suggest you skim through this section. The purpose of this section is to give the user a reasonable model of how this group manipulation system works so that the user has an appropriate framework for thinking about how to use this system to solve his problem.

This system is oriented to interactive use. That means that what the computer does next depends on what you type in next and that what you type next is dependendent to a degree on what the computer has just typed out, i.e. the computer output provides useful information to the user that is used in deciding what to type next.

To provide the computer with input you type it on the terminal and do a carriage return at the end of the line you just typed. The computer does not "read" what you have typed until the carriage return is performed. There is one exception to this rule. If you press the key "CTRL" simultaneously with some other key, the computer automatically sends that one character (the other key pressed) command to the software operating system for processing. The instructions below will explain how to use the ctrl key type of commands that you will need.

The way the computer interprets your input depends on what program system the computer is currently processing. When you logged on, the DDT operating system executed the log on and control remained with it. This operating system understands, among other things, commands that ask it to run various system programs. Also, this is the system that understands the logout command.

When you logged on DDT automatically executed commands found in an initialization file. These commands printed the mail, some general purpose information, loaded the TEN50 operating system, and passed control to it. The TEN50 operating system is used to to execute FORTRAN programs. The CAMAC group manipulation system, herafter referred to as the GROUP system, is written in FORTRAN. This operating system is also used to print files such as this one.

When you type "r group" followed by a carriage return, the GROUP system is loaded and control passes to the GROUP system command processor. The rest of this section will outline the philosophy behind how this system works.

The programs in the GROUP system operate on groups or on pieces of information known about a group such as a coset table over a particular subgroup. The computer stores ail that it knows about a group in a group state table, hereafter referred to as a GST. If a group does not have the proper information in its GST to execute a command, you will get an error message. Most of the commands can be used to add pieces of information about the group to its GST. The PRINT command is used to type out information contained in the GST of a group. The DELET command is used to delete or change items found in the GST of a group. The READ command is used to define a new group. this creates a blank GST and asks you to fill in certain initial pieces of information about the group such as the presentation or generating permutations.

Most of the commands in the GROUP system require a large number of arguments (approx. 6). Instead of typing the command and all of its arguments on one line, the command processor will ask for each piece of information one at a time. At each step the computer types a short description of the piece of information it is expecting now (this is called a prompt.) the computer expects a command as a opposed to an argument, it will type a single colon (:) with no prompt.

The input subroutine in the GROUP system ignores all blanks. A period is always interpreted as meaning the end of the line of input. However no period need be typed as long as the last input character is not a coninuation mark (@). An atsign (@) put at the end of a line of input is interpreted as meaning that you want to continue the input on the next line.

If you are unsure exactly what the computer is expecting, you may type a question mark (?) followed by a carriage return. The computer will respond with a description of the syntax and semantic interpretation of the piece of information it is expecting now. Then it will repeat the same prompt as before and wait for you to provide that piece of information it just described.

If you do not type anything except a carriage return, this is called a default. The prompt and/or the question response will explain how the computer will interpret a default.

Since you will usually want to default on most of the arguments of a command, the computer will first ask you to type in the most important arguments to the command and then will type the prompt "OPTIONS?: ". If you type yes here it will then ask you for each of the remaining arguments to the command. If you default on this prompt, it will set all of the remaining arguments to the default condition and then continue.

If at any time you change your mind about using a command while still typing in the command or the argument list, you can abort, i.e. cause the command processor to forget everything you typed since starting on the current command, by typing a backslash (\) followed by a carriage return. Since the system does not have an error handler capable of restoring the original state of the system, it will type "ERROR SEGMENT ENTERED" as a mark on the printout to show where the computer might have entered an improper state. If after this you get an error message you do not understand, we suggest that you execute a NEWJB command which reinitializes the system. unfortunately this command also causes all stored information about your groups to disappear.

Summary of instructions:

1. The GROUP system is stored in the MATHLAB computer at MIT (at 545 Tech Square, Cambridge). This computer is part of the ARPA net, and the system can be run from any terminal on the net. If you do not have access to the net, you can use the terminals in rm 800D at 545 Tech Square. To log on,

press the keys "CTRL" and "z" simultaneously. Wait for a computer response. Then type ":login group" followed by a carriage return.

2. To start the GROUP system, first check to make sure you are in the TEN50 operating system. If there is a period (.) at the beginning of the current line, you are. If not, check to see what character is at the beginning of the current line. If it is an asterisk (*), you are in the DDT operating system. To get into TEN50, press the "ESC" key (also known as "ALT") followed by a "p", then press the "CTRL" and "c" keys simultaneously. If the character is a colon (:), you are already in the GROUP system; stop worrying. If it is none of these, press the keys "CTRL" and "z" simultaneously. You are then in DDT; follow the above instructions from there.

3. To start the GROUP system from the TEN50 operating system, type "r group" followed by a carriage return. The computer will respond with a reassuring message. When a colon (:) is typed at the beginning of the line, you are ready to give GROUP commands.

4. If you make a typing error, press the rubout key and the computer will "erase" the last typed character. However, since it cannot physically erase the character, it will retype the character it has just "erased" enclosed in slashes (/) to show you what it "erased". If you make a lot of typo's, you can "erase" the entire line by pressing the keys "CTRL" and "u" simultaneously. Do not follow that with a carriage return. Just retype the line.

5. In the GROUP system, all input of any type must be followed by a carriage return. The computer interprets a blank line followed by a carriage return the same as a single period followed by a carriage return. This is called a default. To continue your input on the next line, make an atsign (@) the last character on the current line.

6. You may have a program or subprogram of GROUP commands stored in a file and have the system read them from there. A file name has two parts separated by a blank. The first part of the file name may not have more than five characters. For the system to read that file the second part of the file name must be "dat @". This file should be created using TECO. At any point when you want the system to read your commands from a file, type an atsign (@) at the beginning of the line followed by the first part of the file name followed by a carriage return. The system will continue to read your commands from that file until it reaches the end of the file. After that you must enter your commands from the console in the normal manner. When the system reads commands from a file it ignores all blank lines. It does not interpret them as defaults. Therefore, you may not use defaults when using a file to hold commands.

7. Every GROUP command begins with a string of at most five characters followed by a carriage return. The computer will then guide you through the argument list for that command.

8. At any point where the computer is waiting for you to tell it something, if you type a question mark (?) followed by a carriage return, the computer will give you information about what it is expecting: both the syntax rules and the semantic interpretation of what you are about to type.

9. At any point when the computer is waiting for you to tell it something you may abort a command by typing a backslash (\) followed by a carriage return. The backslash always has this special meaning. you cannot use it for any other purpose (like in a group name).

10. Here is a summary of the commands:

| | |
|---|---|
| READ | Input command to define a group |
| PRINT | Print information that is known about the group |
| DELET | Delete a group or item of information about the group |
| NEWJB | Initialize system (all stored data lost) |
| GNGRP | Generate a permutation group |
| TDOCX | Create coset table or coset representatives (applicable only for group with presentation) |
| COSET | Find coset table for arbitrary group and subgroup |
| RDSCH | Find presentation for a subgroup |
| RELAT | Find presentation for a permutation type group |
| SUBGP | Find generating permutations for a group |
| NRMZR | Create normalizer of permutation type subgroup |
| CNTZR | Create centralizer of perm type subgroup |
| STABL | Create stabilizer subgroup for a group around a point |
| ORBIT | Find orbits for a perm type group |
| BLOCK | Complicated, see Tech. Report No. 8 |
| V-ORB | Find orbits of vectors with respect to a permutation group |
| NORML | Says if subgroup is normal (presentation required) |
| CVORB | More general v-orb that also gives weight distibutions |

11. The system is not yet completely debugged and it does not completely protect you from your own mistakes. If you get an error message that includes the word exit, you will no longer be in the GROUP system, but in TEN50 instead. To restart the GROUP system without loss of data, type "start" followed by a carriage return. then follow the instructions the system gives.

12. If you want to stop the computer while it is printing out something long (like a coset table) press the keys "CTRL" and "c" simultaneously until the computer stops printing and responds with a period. To re-enter the GROUP system type "start" followed by a carriage return.

13. If you think you noticed a bug in the GROUP system, please tell us. Phone Vera Pless or Randell Weiss at x3-6026 (545 Tech Square, rm. 830). Or if you are knowledgeable about the ITS system, send mail to user RBW. Thank you.