Appendix B

Glossary for Groups32

#Cmds

(-- num) Value

VALUE - Number of commands currently in use

#Groups

(-- 144) Constant

Number of stored groups

#Set

(set -- #ele) Code

Returns number of elements in set

#SubGroups

Colon

Return the number of subgroups in current list. (Must generate subgroup list first).

$!

($ addr --) Colon

Store counted string $ at addr

$.

($ --) Colon

Print counted string

$<

($1 $2 -- t/f) Colon

Compare counted strings - True if $1 < $2

$=

($1 $2 -- t/f) Colon

Compare counted strings - True if $1 = $2

$Array

(<$sz><arrsz> --) Colon

Syntax <stringsize><arraysize> $Array <name> This creates an array of strings. If X is the array then <n> X returns the address of the nth string. Indexed 0

$Compare

($1 $2 -- -1 | 0 | 1) Colon

Compare counted strings. Return -1 if $1 < $2, 0 if $1=$2, 1 if $1 > $2 -- lexicographic order

$Pfile

(addr cnt) Colon

This starts a file for redirecting output (see Pfile) but it takes as its argument the filename provided as a string in the form of address and count.
e.g. s" temp.out" $Pfile will start a file temp.out

(Ins)

(n i --) Colon

Insert n at position I in current list -- other entries moved up. NOTE: this assumes that n is not in the list and that I is the proper place to insert. See OL-Insert.

(IsIn?)

(n lst -- i) Colon

i is either the position where n is found or the position where it should be inserted.

.Ele

(ele --) Colon

Print element (using A, B, ...) followed by a space

.Ele_

(ele --) Colon

Print element (using A, B, ...) followed by underscore

.LID

(--) Colon

Print the name of the current list

>CMD

<name> <action> Colon

Introduce new command to current tree. The syntax is >CMD <command name><execution word> . The <command name> is what is to appear on the menu, the <execution word> is the word to be executed if the command is selected

>Elsewhere

(exec --) Colon

This installs the execution address of a routine to search for a storage address "elsewhere" (other than the parameter stack) in the current temp-storage structure. The installed routine should have the stack diagram (addr -- found?).

>Group

(grp# --) Colon

Install group given by grp# as current group. Set Gord Gnum, Grp, and Set-G.

>Lst

(list --) Colon

Install a list as the current list

2^n

(n -- 2^n) Code

Raise 2 to the input power. Also used to produce singleton set (see Singleton).

2Carray

<#rows><#cols> 2Carray <name> Colon

This is a defining word used to create 2 dimensional arrays of bytes. 3 5 2Carray XXX creates an array XXX. 1 2 XXX returns the address of the entry in the first row and 2nd column of the array XXX.

Address

(indx -- addr) Colon

If a Temp-Storage array is created and made the current storage structure, n Address is the address of the nth temporary storage location. Address is primarily used to initialize the addresses.

Array

<size> Array <name> Colon

A defining word for creating arrays of integers. 5 Array XX creates an array XX indexed 0..5. 3 XX returns the address of the integer with index 3.

ASK

(n --) Colon

Get up to n characters from the keyboard and store them in the TIB (terminal input buffer). This is one of the ways that a program can ask for keyboard input and ,ake the system regard it as interpreted input.

AT

(x y --) Colon

Set Cursor position -- upper left is (0,0)

AT?

(-- x y) Colon

Return cursor position -- upper left is (0,0)

C.

(perm --) Colon

Print permutation as product of disjoint cycles.

Calc-Orders

(--) Colon

Calculate the order of each element in the current group and the number of elements of that order

Center

(-- center) Colon

Returns center of current group

Centralizer

(ele -- centrz) Colon

Returns centralizer of ele

CLRLN

(row --) Colon

Clear line given by <row>

CMD?

<command name> Colon

This types the name of the execution word associated with the menu command (used in command completion interface).

CmdSize

(-- 18) Constant

CONSTANT - max size of name of command

Commands

Colon

This initiates the command completion interface. It is a loop which gets commands and executes them. It supplies the prompts for the main and permgrps trees

Conj-Classes

(grp# --) Colon

Print conjugacy classes for given group.

Contains

(set1 set2 -- f) Code

Returns true if set1 contains set2.

Cosets

(subg --) Colon

Chart of left/right cosets of subg generated by set

Defaults

(restore saved 1-5) Colon

Use after Install to restore original tables 1-5

EEOL

Colon

Clear current line

EleOrder

(ele -- ord) Colon

Return order of element in current group

Empty-Set

(-- 0) Constant

Returns the empty set

EnterData

(--) Colon

Used in Search-Grp and Search-All to prompt for entry of generators and relations.

EntrySize

(-- 22) Constant

CONSTANT - size of name field + execution address

EVAL

EVAL < string in group_letters> Colon

Takes following string in group letters with x' indicating inverse of x. Returns product in current group.

Examples

Colon

This imports tables for a selection of examples in the file examples.f

For-All-Eleme

(-- hi lo) Colon

Put limits on stack for loop over all elements of current group

For-All-Group

(-- hi lo) Colon

Put limits on stack for loop over all groups

For-All-Prope

(-- hi lo) Colon

Put limits on stack for loop over list of subgroups (must generate subgroups first). USE ?DO rather than DO.

For-All-SubGr

(-- hi lo) Colon

Put limits on stack for loop over list of subgroups of current group (must generate subgroups first)

G#Range

(-- low hi) Colon

Currently puts 1, 144 on stack -- used for checking if input represents a valid group number.

G*

(i j -- i*j) Code

Multiplication in current group

G32Version

Colon

Print current version of Groups32

Generates

(set -- subg) Colon

Returns subgroup of current group generated by set

Generate-SubG

(grp# --) Colon

Generate but do not show the list of subgroups for grp#

Generators

(n -- gens or empty) Colon

Return minimal set of generators for n-th subgroup (as bitmap) in current subgroup list. The command Generate-Subgroups or Subgroups must be used to generate the list first.

Get-Ele

(-- ele) Colon

Prompt for element -- wait until element is in range for current group

Get-Ele.

(-- ele) Colon

Gets the element and echos it

Get-Element

(-- ele) Colon

Repeat wait for key press until valid (in group) - return element

Get-Grp

(-- n) Colon

Prompt for a group -- wait for input of valid group number

Get-Key

(-- ele) Colon

Wait for key press, return element

Get-Num

(-- n) Colon

Prompt for a number -- wait for input of valid number

Get-Set

(-- set) Colon

Prompt for a set -- checks that elements are in range for current group

Get-TIB

(--) Colon

Read a line of input into the Terminal Input Buffer

Gnum

(-- gnumber) Value

VALUE - Returns number of current group

G-OK

(n -- n t | f) Colon

Check if n is a valid group number (see G#Range)

Gord

(-- gorder) Value

VALUE - Returns order of current group

GroupData

(-- addr) Create

The starting address of the block of memory where tables are stored.

Grp

(-- [gaddr]) Create

VARIABLE - containing base address of table

Help:

help text terminated by Help; Colon

Within a definition this starts a help-comment that is printed when the user types X or INFO before a command. The text should begin on a new line and it should end with Help; on a new line

ID

(-- 'A') Constant

ASCII value used for the identity (internally it is 0 -- which is printed as "A" -- ASCII value is 65)

Ident

(-- 0) Constant

The internal representation for the identity element

Idx

(-- addr) Create

The starting address of the block of memory where orders of the groups are stored

Install

(n --) Colon

Install group, currently in the position of group 0, and move it to take place of group n. N must be 1-5.

Inv

(ele -- ele') Colon

Return inverse of ele in current group

IsIn?

(n list -- i flag) Colon

If n is an entry in list, this returns position of n and a true flag. If n is not an entry it returns the position where n should be inserted and a false flag.

Isomorphic

(grp1 grp2 --) Colon

This allows the user to map elements of grp1 to grp2. It automatically supplies consequences of a map. It prompts when an element is mapped illegally.

ISOWindow

Colon

Set the system so that isomorphisms are handled in a separate pop-up window (this starts a second copy of the program and requires that the two copies communicate by disk files.)

LCmp

(i j -- f=-1,0,1) Colon

Compare data at I with data at j in current list

LCoset

(ele subg -- lcos) Colon

Returns left coset of ele with respect to subg

Left->Right

Colon

Set the system to muliply permutations from left to right.

LMax

(-- m) Colon

Size of list (defined as n when created)

Lst

(i -- addr) Colon

Address of ith entry of current list

'Lst

(-- addr) Value

VALUE -- address of current ordered list

Lst!

(n i --) Colon

Store index n to be ith entry

Lst.

(--) Colon

Print the current list (as list of indices)

Lst@

(i -- entry) Colon

ith entry of current list

Max#Cmds

(-- 30) Constant

CONSTANT - maximum number of commands in interface

MaxOrd

(-- 32) Constant

Largest group order (32)

MaxSubs

(-- 200) Constant

Maximum number of stored subgroups

MaxTables

(-- 150) Constant

Maximum number of tables

Member!

(ele set -- set') Code

Make ele a member of set

Member?

(ele set -- f) Code

Returns true if ele is a member of set

NCompare

($1 $2 n -- -1 | 0 | 1) Colon

Compare first n characters

NO-ISOWindow

Colon

Do not handle isomorphisms in a separate pop-up window.

Normal?

(subg -- f) Colon

True if subg generated by set is normal

Normalizer

(subg -- normal) Colon

Returns the normalizer of given subgroup

OL-Copy

(Lst1 Lst2 --) Colon

Copy the data from Lst1 to Lst2

OL-Delete

(i lst --) Colon

Remove the element at position I -- move everything else down.

OL-Empty

(lst --) Colon

Empty the list.

OL-Empty?

(lst -- f) Colon

Is the given list empty?

OL-Insert

(n list --) Colon

Insert n in the list at the proper position (by order) -- this allows duplicates

OL-Print

(lst --) Colon

Print the given list (makes it the current list)

Orders

(grp# --) Colon

Makes group current and lists elements by order.

OrdList

< size> OrdList <list_name><comp> Colon

Defining word for ordered lists -- requires a comparison operator on indices: comp(I,j) is -1 if L[I]<L[j], 0 if = and 1 if >. A list saves 1-byte indices to data stored elsewhere as an array structure.

P*

(perm1 perm2 -- result) Colon

Permutations are stored as counted strings: a count byte followed by the images of 1,2,3,.... This returns the product evaluated either Left->Right or Right->Left (see these words).

P.

(perm --) Colon

Print the permutation in "long form" i.e. the images of 1,2,3,...

P/

(perm1 perm2 -- res) Colon

This computes perm1 * (perm2)^-1 if permutations are multiplied left->right and (perm2)^-1 * perm1 if permutations are multiplied right->left.

Pclose

(--) Colon

Close a printing file to which output has been redirected (see Pfile).

Perm:

(-- addr) Colon

Input a product of disjoint cycles using single parentheses [e.g. Perm: (1 2)(3 4 5)] Returns the address of the counted string that represents the permutation. In the above example the string is 5 2 1 4 5 3.

PFecho

(-- addr) Create

If output is being redirected to a file (see Pfile) this determines if output is also echoed to the screen (PFecho ON) or not (PFecho OFF).

Pfile

Pfile <filename> Colon

This opens a file for redirection of output (e.g. Pfile Temp.Out opens the file Temp.Out for writing). What gets sent to the new file depends on the variable Printing. It can also be controlled by the command Print.

PInvert

(perm -- inv) Colon

Compute perm^-1.

Print

Print <command line> Colon

Execute and print to a file the following commands (must open a file for printing first, see Pfile and Pclose)

Printing

(-- addr) Create

If a file is open for printing (see Pfile and Pclose) Printing ON will direct subsequent output to the file until Printing OFF.

Prt-Orders

(--) Colon

Print the elements of the current group sorted by order

PSize

(-- n) Value

Current n so that permutations are in Sn. This is a VALUE which is set by TO (e.g. 3 TO PSize)

PTemp

(-- addr) Colon

Return the address of the next available temporary storage location for permutations.

RCoset

(ele subg -- rcos) Colon

Returns right coset of ele with respect to subg

ReadGroup

Colon

Allows importing tables using a file. The file has ReadGroup on top line, symbols one per line, a blank line, then the table (one row per line), and finally \s on the next line.

Right->Left

(--) Colon

Set so that a product of cycles is multiplied from right to left. (1 2)(1 3) = (1 3 2)

Search-All

(--) Colon

Assume generators and relations have been entered (see EnterData) this initiates a search of all groups.

Search-Grp

(grp# -- found?) Colon

Assume generators and relations have been entered (see EnterData) this checks if given group satisfies conditions.

Set-

(sub1 sub2 -- sub1-sub2) Colon

Returns difference of sets

Set&

(sub1 sub2 -- intersection) Colon

Returns intersection of sets

Set*

(set1 set2 -- product) Code

Returns the set of products of elements in set1 set2 in given group.

Set.

(set --) Colon

Prints set (given as integer bitmap) using letters for elements

Set+

(sub1 sub2 -- union) Colon

Returns union of sets

Set-G

(-- gset) Value

VALUE - Returns underlying set of current group

Set-Insert

(n list -- t/f) Colon

This inserts n in the list at the proper place provided it is not already in the list -- no duplicates.

Show-SubGroup

(--) Colon

Print current list of subgroups (must be generated first by Generate-Subgroups)

Singleton

(n -- {n}) Code

Returns set with ele as only member

SubGroup

(n -- subg or empty) Colon

Return n-th subgroup (as bitmap) in current subgroup list. The command Generate-Subgroups or Subgroups must be used to generate the list first.

SubGroups

(grp# --) Colon

Makes group current and lists all subgroups with generators.

SubGrp.

(subg -- ;;; print subg) Colon

Prints subgroup (bitmap as set) using letters. This is the same as Set.

SUBGtable

(-- addr) Create

Base of record structure which holds subgroups of current group and generators for them

TAB

(n --) Colon

Tab to next position multiple of n

Table

(n --) Colon

Makes group current and prints multiplication table.

Table_Size

(-- 1024) Constant

Size of stored tables in bytes

Temp

(-- addr) Colon

Print the address of next available storage location in the current temp-storage structure. (A storage structure is made current by using its name -- e.g. LISTS TEMP returns the next address in the LISTS structure.

Temp-Storage

(marks) Colon

Defining word. Temp-Storage XXX creates a structure named XXX with slots for addresses. The addresses must be allocated separately and filled in. Also a procedure to execute if active addresses are removed from the stack. See >Elsewhere and Address.

Top

(-- top) Colon

1+ position of last index in current ordered list (part of Ordered List module).

Translation

string Colon

Store string in current translation table. Example: if group is of order 5, Translation BCDAE will print table 5 with A ->B, B->C, C->D, D->A, A->E

Tree1

(--) Colon

Set system so that following commands are put in the main menu

Tree2

(--) Colon

Set system so that following commands are put in Permgrps menu

TTab1e

(n --) Colon

This prints the table obtained from table n by applying a translation of elements - see Translation

Y/N?

(-- t/f) Colon

Wait for key input of y (returns TRUE) or n (returns FALSE). Case insensitive.