

GAP Examples and Exercises 1.

This material is for the exercise classes in the afternoons following the tutorials. The idea is to try some things which are not available in GAP demo mode, and to get used to the basics. Optional extras at the end.

Starting and stopping GAP; Help; Logging; Suppressing printing; Line editing; Command history.

Log in to one of the lab machines. Open a terminal and, at the UNIX prompt, type

```
module load gap
```

followed by

```
gap
```

Then type

```
quit;
```

followed by

```
gap -b
```

to suppress the banner.

Now try

```
gap> SaveWorkspace( "test.ws" );
```

Then quit again and restart with

```
gap -L test.ws
```

Notice how much faster GAP starts in this way. Of course you can also save a workspace in which you have defined your own user variables, in order to be able to return to your session and its objects later. Such workspaces can be big.

The rest should be done at the GAP prompt. Notice how the following navigates around a help topic.

```
gap> ?list
gap> ?4
gap> ?>
gap> ?
gap> ?<
gap> ?c 1 o
gap> ?1
```

Find out about Line editing from the tutorial using the built-in help.

From this point, try logging your session, with (eg)

```
gap> LogTo("test.log");
```

Later you can review your session by examining this log file at the UNIX prompt (use `cat`, or `more`, or a text editor). If you want to stop logging before your session ends, you can - find out how using the help system!

Eg. Saving an object:

```
gap> s := SymmetricGroup( 10 );
gap> c := Normalizer( s, Subgroup( s, [ (1,2,3)(4,5,6,7,8)
(9,10) ] ) );
gap> Size( last );
gap> PrintTo( "cent.g", "c := ", c, ";\n" );
gap> Unbind( c );
gap> IsBound( c );
gap> Read( "cent.g" );
gap> Size( c );
```

Eg. Suppressing printing:

```
gap> els := Elements( AlternatingGroup(5) );;
gap> Length( els );
gap> Number( els, perm -> OrderPerm( perm ) = 2 );
```

Next type in the following command, but try to use tab-completion for the first function.

```
gap> Filtered( [1..100], x -> IsPrime(x) );
```

How many letters must you type before `Filtered` (partially) completes? What happens if you double-tab part way through the word (after `Filter`, say)?

Use the cursor (arrow) keys to retrieve your command and edit it. Assign a name, say `l`, to the list somehow. Then check

```
gap> Length(l);
```

Try constructing other lists using the functions described in the Lists and Records section of the tutorial.

Try these after having been in an active session for a while:

Go back to a command you typed earlier. Execute it again, then type the down arrow. What happens?

Choose the initial letter of a command you typed earlier in the session. Type that letter, then the up arrow (perhaps more than once). What happens?

Elementary GAP programming

Here is a GAP function

```
gap> oddSquares := function( n )
>   # Sum up the odd squares of numbers up to n
>   local i, sum;
>   sum := 0;
>   for i in [1..n] do
>     if ( i mod 2 = 1 ) then
>       sum := sum + i^2
>     fi;
>   od;
>   return sum;
> end;
```

Note how comments work and where semicolons are used (and not used).

Enter(*) this function into a running GAP and execute it for some values of n . Can you load this function into GAP by putting it in a text file (without the prompts) and Reading that file?

(* You can cut-and-paste into an interactive session, even with the prompts.)

You have seen the `Display` function applied to matrices. Write an equivalent function `displayMat` which produces similar 'nicely' formatted output that GAP is able to read as input. Test your function.

All too easy?

If you are happy to start work with groups, try

<http://www-circa.mcs.st-andrews.ac.uk/CIRCA/WkShopTalks/SL/lab.pdf>

For some advanced (general) examples see

<http://www.gap-system.org/Doc/Examples/examples.html>

And for advanced(!) projects in GAP see

<http://www.math.colostate.edu/~hulpke/gaproj/projects.htm>

John McDermott, RISC Summer School 2008.