

-1 Chapter: An Introduction to GAP

This chapter provides the instructions needed to use this manual as well as the very basics on using GAP. Much of the material in this chapter is based on material written by Russell Blyth at Saint Louis University. GAP is a free software program. This manual assumes you have installed version 4.4. If you do not have this version already, you can download a copy from:

<http://www.gap-system.org/>

A reference manual and extensive tutorial for the software GAP are also available at this web site.

Some Commands That You Will Use Often

- 1) To exit from GAP type `quit`;
- 2) To save your work in a file type `LogTo("filename");`
- 3) To stop saving your work in that file type `LogTo();`
- 4) Type `<ctl>-p` to redisplay the previous command
- 5) If an error causes GAP to enter a loop (and gives you the prompt `brk>`) you can exit the loop by typing `<ctl>-D` or `quit`;
- 6) Type `?` followed by a subject name to get help about that subject.

The `?` command, listed as (6) above, is particularly useful. This can be used whenever you forget the exact command for something or when you wonder if GAP has a command for a particular operation. For example, suppose you want to know how to denote multiplication in GAP. Type

```
gap> ?multiplication
```

and GAP returns a description of how to enter basic arithmetic commands including the following:

```
'<a> * <b>' denotes the multiplication of multiplicative elements <a> and <b>.
```

Careful: GAP distinguishes between upper and lower case letters. (So, for example, `LogTo` is not the same as `Logto`.)

Careful: Always put a semicolon at the end of a command. Two semicolons at the end of a command will cause GAP to execute the command but not list the output of the command.

Getting Use to GAP

We will now do some exercises to get you comfortable with GAP.

At the gap prompt type `LogTo("assignment-1");` This will save the following work in a file called "assignment-1". At the gap prompt type `(5 + 3) * 9;` then hit the enter key. Your screen should look like this:

```
gap> LogTo("assignment-1");
gap> (5 + 3) * 9;
```

72

At the prompt type $(5 + 3) * 9$ (without the semicolon) and then hit enter.

```
gap> (5 + 3) * 9  
>
```

Notice nothing happens. **GAP** does not know you have finished the command because there is no semicolon. Now type `;` and then hit return. You should now get the 72.

```
gap> (5 + 3) * 9  
>;  
72  
gap>
```

Now type $(5 + 3 * 9;$ and hit enter. Your screen should look like

```
gap> (5 + 3 * 9;  
Syntax error: ) expected
```

An error message occurred because the number of left parentheses and right parentheses does not match. Type `<ctl>-p` to redisplay the previous command. Then use your arrow keys to insert the needed parenthesis.

```
gap> (5 + 3 * 9;  
Syntax error: ) expected  
gap> (5 + 3) * 9;  
72  
gap>
```

GAP can be used to test the equality of two values. At the **GAP** prompt type $6=9;$ and then hit enter.

```
gap> 6=9;  
false
```

GAP returned the value of false since 6 is not equal to 9. Now use **GAP** to complete the following exercises. Don't forget to put a semicolon at the end of a command!

- 1.1 Compute 3^{121} .
- 1.2 Determine if $2^{25} + (45 * 51,777)$ is greater than 34 million.

You can assign values to variables in **GAP** by using `:=`. This allows you to refer to an object with a name. The name of the variable is called an identifier. In the following example `a` is an identifier.

```
gap> a := (10 + 7) * (9 - 6);
51
gap> a;
51
gap> a * (a-1);
2250
gap> a:=14;;
gap> a * (a-1);
182
```

Notice when an identifier is assigned a value that value is echoed on the next line. Because two semicolons occurred in the line `a:= 14;;` the value of `a` was not echoed. Almost any sequence of letters and digits containing at least one letter can be used as an identifier. Use **GAP** to complete the following exercises.

- 1.3 Assign the value 4 to the identifier `b`.
- 1.4 Assign the value 522456 to the identifier `bignumber`.
- 1.5 Compute `b + bignumber`.

This completes the computer exercises for Chapter -1. At the gap prompt type `LogTo()`; . This causes **GAP** to stop saving your work to the file “assignment-1”. Exit **GAP** by typing `quit`; at the gap prompt. Then print the file “assignment-1”.