

Introduction to GAP

The 7th Biennial International Group Theory Conference
(7BIGTC2023) at North-West University, South Africa

C. S. Anabanti



8–10 August, 2023

AGENDA

- ▶ Day 1: August 8, 2023 (Tuesday)
 - ▶ 14:00–14:15: Pre-attendance GAP workshop check: installation, starting, how to save your work and load files.
 - ▶ 14:15–14:45: Introduction to GAP (Part I)
 - ▶ 14:45–15:00: Exercise session I
- ▶ Day 2: August 9, 2023 (Wednesday)
 - ▶ 14:00–14:40: Introduction to GAP (Part II)
 - ▶ 14:40–15:00: Exercise session II
- ▶ Day 3: August 10, 2023 (Thursday)
 - ▶ 08:00–09:00: Introduction to GAP (Part III)
 - ▶ 09:00–09:25: Exercise session III
 - ▶ 09:25–09:55: Project activities
 - ▶ 09:55–10:00: Concluding remarks

Introduction to GAP

GAP: Installation, starting, how to save your work and load files

GAP as a fancy calculator

Functions in GAP: List, Groups, Help function et-cetera

Possible error messages and how to fix them

Exercise I

Small Group Library in GAP

Testing whether two groups are isomorphic

Cayley's theorem and rewriting a group as a permutation group

Finitely presented (f.p) groups in GAP

Other algebraic systems

Exercise II

Regression tests in GAP

Defining functions in GAP

Class Exercise

Exercise III

Project activities

Concluding remarks

GAP: Installation, starting, how to save your work and load files

14:00 – 14:15 on August 8, 2023

- ▶ Installing GAP: Windows, MAC and Linux users
- ▶ Starting and leaving GAP
- ▶ How to create a file that will contain your GAP outputs
- ▶ How to load a GAP source code from a file
- ▶ Some of the things GAP helped me do!

Installing GAP: Windows, MAC and Linux users

- ▶ For Windows users: An easy way would be to download gap-4.12.2-x86_64.exe file (578MB) which can be downloaded from <https://www.gap-system.org/Releases/4.12.2.html>.
- ▶ For MAC users: A good solution for Mac would be Gap.app (see the link at <https://www.gap-system.org/Download/alternatives.html>) which is precompiled for macOS.
- ▶ For Linux users: Downloading tar.gz or zip archive from <https://www.gap-system.org/Releases/4.12.2.html> may work, if you have C compiler and some other tools preinstalled (if not, and there is at least some internet connection, it may still be helpful to download GAP tar.gz or zip archive to reduce downloads).

The latest version of GAP is Version 4.12.2, and it was released in December 2022. First-time GAP users may install this version.

Starting and Leaving GAP

- ▶ You can start GAP by typing "gap" at the command prompt/terminal of your operating system followed by the Return/Enter key. Windows users can click on the GAP-v4.12.2 App installed on their computer to start GAP straightaway.
- ▶ We usually close the GAP prompt/terminal by typing "quit;" or "Ctrl-D". You can also achieve the same result by closing the terminal the usual way you close a window/tab of a browser.

Note: To execute any GAP command, you need to press the return/enter key after inserting the command. Below is what will be displayed on your prompt/terminal when you start GAP.

```
GAP GAP 4.12.2 built on 2022-12-19 10:30:03+0000
https://www.gap-system.org
Architecture: x86_64-pc-cygwin-default64-kv8
Configuration: gmp 6.2.1, GASMAN, readline
Loading the library and packages ...
Packages: AClib 1.3.2, Alnuth 3.2.1, AtlasRep 2.1.6, AutPGrp 1.11, Browse
1.8.19, CaratInterface 2.3.4,
CRISP 1.4.6, Cryst 4.1.25, CrystCat 1.1.10, CTbLib 1.3.4,
curlInterface 2.3.1, FactInt 1.6.3, FGA 1.4.0,
Forms 1.2.9, GAPDoc 1.6.6, genss 1.6.8, IO 4.8.0, IRREDSOL 1.4.4,
LAGUNA 3.9.5, orb 4.9.0,
Polenta 1.3.10, Polycyclic 2.16, PrimGrp 3.4.3, RadiRoot 2.9, recog
1.4.2, ResClasses 4.7.3,
SmallGrp 1.5.1, Sophus 1.27, SpinSym 1.5.2, TomLib 1.2.9, TransGrp
3.6.3, utils 0.81
Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap>
```

How to create a file that will contain your GAP outputs

Suppose you want to look at what you would do today in GAP at some point in the future, you may consider saving your GAP outputs to a new file. You can use the command:

```
LogTo("August.g");
```

to tell GAP to start saving your work to the new file named "August.g". When you want to stop saving to that file, type the command:

```
LogTo();
```

to tell GAP to stop saving in the existing file named August.g.

Warning! All the GAP outputs before typing 'LogTo("August.g");' and after typing 'LogTo();' will not be saved.

Some remarks

1. To log into a file, you need to determine a filename. So you may choose, for example, today's date as your filename for ease of remembrance. You type the command `LogTo("8August2023.g");` for GAP to create a file called `8August2023.g` in your home directory. From then henceforth, anything you do will be saved in the file `"8August2023.g"`.
2. To log out of a file, you simply type `LogTo();` in the GAP terminal, and GAP will automatically log you out of the file. After you have been logged out of a file in GAP, you can either create a new filename for saving your work after the logout or continue on the terminal without saving your work at all.
3. Suppose you are already logged into the file `"8August2023.g"`, and you remember this after some computations. You may try logging in again to check. To do so, you type `LogTo("8August2023.g");`, and GAP will display `"Already logging to 8August2023.g"` as the output...that shows that you are logged in already! Alternatively, you may type `LogTo` and then click the down arrow key once to see the last `LogTo` option you used.

Note that if you log out of a file and login again with the same filename, then you will overwrite the initial file.

How to load source code from a file

Suppose you have a file called "Pretoria.g" which contains some GAP codes which you want to use in your GAP computations. Instead of typing all the GAP codes one after the other, you can just load this file into GAP using the Read command, as follows:

```
Read("Pretoria.g");
```

Note that after you have called the above GAP Read code, GAP will automatically see everything in the file "Pretoria.g" as if you have typed all of them into the GAP prompt.

Exercise

We gave you a file called BIGTC2023.g. We want you to do the following.

1. Save BIGTC2023.g in a folder where your installed GAP can see it.
2. Start GAP and do (a) to (c) given below.
 - (a) Create a file called "PreBIGTC.g".
 - (b) Read the file "BIGTC2023.g" given to you.
 - (c) Log out of your PreBIGTC.g file.

Some of the things GAP helped me do!

- ▶ Anabanti CS, Hammer S and Okoli NC: An infinitude of counterexamples to *Herzog's* conjecture on involutions in simple groups. *Communications in Algebra* 49(4)(2021), 1415–1421.
- ▶ Anabanti CS: On the three questions of *Bertram* on locally maximal sum-free sets. *Quaestiones Mathematicae* 44(3)(2021), 301–305.
- ▶ Anabanti CS: A question of *Malinowska* on sizes of finite nonabelian simple groups in relation to involution sizes. *C. R. Math. Acad. Sci. Paris* 358(11-12)(2020), 1135–1138.
- ▶ Anabanti CS: A counterexample to *Zarrin's* conjecture on sizes of finite nonabelian simple groups in relation to involution sizes. *Archiv der Mathematik* 112(3)(2019), 225–226.
- ▶ Anabanti CS: Three questions of *Bertram* on locally maximal sum-free sets. *Applicable Algebra in Engineering, Communication and Computing* 30(2)(2019), 127–134.
- ▶ Anabanti CS and Hart SB: On a conjecture of Street and Whitehead on locally maximal product-free sets. *Australasian Journal of Combinatorics* 63(3)(2015), 385–398.

Introduction to GAP (Part I)

14:15 – 14:45 on August 8, 2023

Basic operations in GAP

```
gap> 2+3;
```

```
5
```

```
gap> 2-3;
```

```
-1
```

```
gap> 2*3;
```

```
6
```

```
gap> 2/3;
```

```
2/3
```

```
gap> Float(2/3);
```

```
0.666667
```

```
gap> Rat(0.25);
```

```
1/4
```

```
gap> 2*(3-5+7);
```

```
10
```

```
gap> 5 mod 3;
```

```
2
```

Functions in GAP

- ▶ A program written in the GAP language is called a function.
- ▶ Functions are special GAP objects. Most of them behave like mathematical functions. They are applied to objects and will return a new object depending on the input.
- ▶ Example: The function 'Factorial' returns the factorial of an integer; see below for illustrations.

```
gap> Factorial(4);
```

```
24
```

```
gap> Factorial(6);
```

```
720
```

```
gap> Binomial(5,2);
```

```
10
```

```
gap> Binomial(5,3);
```

```
10
```

The Help function in GAP

What I term the most important function in GAP is the help function. To ask GAP how to use the function 'dihedral', simply type "?Dihedral" or "?DIHEDRAL" or "?dihedral" or "?dihed" and click the Enter key. You must not spell it out in full...you can also use the first few letters such as "?dih", "?dihe", "?dihed" et-cetera. One take home here is that we need: **?WhatYouNeedHelpAbout**.

```
gap> ?Dihedral
Help: several entries match this topic - type ?2
to get match [2]
[1] Reference: DihedralGroup
[2] CTblLib: dihedral groups!character table
[3] LAGUNA: DihedralDepth
[4] polycyclic: DihedralPcpGroup
```

Suppose we want to know more about dihedral groups in GAP, we can then type "?1" and click the enter button.

Another way of asking for help in GAP

There is another way of calling help in GAP; other than the “?WhatYouNeedHelpAbout” discussed earlier. To ask GAP how to use the function ‘Dihedral’, you may use any of
“HELP(“Dihedral”);”, “HELP(“dih”);”, “HELP(“dihe”);”,
“HELP(“dihed”);”, “HELP(“dihedral”);”, “HELP(“DIHEDRAL”);”
et-cetera and click the Enter key. An example is given below.

```
gap> HELP("Dihedral");
```

You can also use any of the options below.

```
gap> HELP("DIHEDRAL");
```

```
gap> HELP("dihedral");
```

```
gap> HELP("dihed");
```

On some functions in GAP: List in GAP (1)

We can determine the element in a certain position in a list.

```
gap> B:=[2,11,3,8,1,10,20,4,22];  
[ 2, 11, 3, 8, 1, 10, 20, 4, 22 ]  
gap> B[2];  
11  
gap> B[4];  
8
```

We can arrange a list of rational numbers sequentially.

```
gap> Sort(B);  
gap> B;  
[ 1, 2, 3, 4, 8, 10, 11, 20, 22 ]  
gap> B[2];  
2  
gap> B[4];  
4
```


On some functions in GAP: List in GAP (2)

```
gap> L:=[1,2,3,4,5];;
gap> Add(L,6);
gap> L;
[ 1, 2, 3, 4, 5, 6 ]
gap> Remove(L,2);
2
gap> L;
[ 1, 3, 4, 5, 6 ]
gap> L[2]=L[4];
false
gap> M:=[1,2,2,2,5,6];
[ 1, 2, 2, 2, 5, 6 ]
gap> M[2]=M[4];
true
gap> Set(M);
[ 1, 2, 5, 6 ]
```

On some functions in GAP: List in GAP (3)

“Difference” and “SubtractSet” are not the same.

```
gap> L:=[1,3,4,5,6];;
```

```
gap> M:=[1,2,3,4,5];;
```

```
gap> Difference(L,M);
```

```
[ 6 ]
```

```
gap> L;
```

```
[ 1, 3, 4, 5, 6 ]
```

```
gap> L:=[1,3,4,5,6];;
```

```
gap> M:=[1,2,3,4,5];;
```

```
gap> SubtractSet(L,M);
```

```
gap> L;
```

```
[ 6 ]
```

On some functions in GAP: List in GAP (4)

The functions "Union" and "UniteSet" are not the same.

```
gap> A:=[1,2,3,4,5];;
gap> B:=[4,5,9,10];;
gap> Union(A,B);
[ 1, 2, 3, 4, 5, 9, 10 ]
gap> A;
[ 1, 2, 3, 4, 5 ]
gap> B;
[ 4, 5, 9, 10 ]
```

```
gap> A:=[1,2,3,4,5];;
gap> B:=[4,5,9,10];;
gap> UniteSet(A,B);
gap> A;
[ 1, 2, 3, 4, 5, 9, 10 ]
gap> B;
[ 4, 5, 9, 10 ]
```

On some functions in GAP: List in GAP (5)

In GAP, every set is a list but not every list is a set.

```
gap> A:=[1,2,3,4,4,5];;
```

```
gap> IsSet(A);
```

```
false
```

```
gap> IsList(A);
```

```
true
```

```
gap> B:=[1,2,3,4,5,6];;
```

```
gap> IsSet(B);
```

```
true
```

```
gap> IsList(B);
```

```
true
```

You can convert a list of integers to a set.

```
gap> C:=Set(A);;
```

```
gap> A;
```

```
[ 1, 2, 3, 4, 4, 5 ]
```

```
gap> C;
```

```
[ 1, 2, 3, 4, 5 ]
```

On some functions in GAP: List in GAP (6)

We can determine the position of an element in a set

```
gap> B:=[1,2,3,4,8,10,11,20,22];  
[ 1, 2, 3, 4, 8, 10, 11, 20, 22 ]
```

```
gap> 5 in B;  
false
```

```
gap> 10 in B;  
true
```

```
gap> Size(B);  
9
```

```
gap> PositionCanonical(B,8);  
5
```

```
gap> PositionCanonical(B,11);  
7
```

On some functions in GAP: List in GAP (7)

```
gap> A:=[1,2,3,4,5];; B:=[4,5,6,7];;
```

```
gap> A-B;
```

```
[ -3, -3, -3, -3, 5 ]
```

```
gap> B-A;
```

```
[ 3, 3, 3, 3, -5 ]
```

```
gap> A+B;
```

```
[ 5, 7, 9, 11, 5 ]
```

```
gap> 2+A;
```

```
[ 3, 4, 5, 6, 7 ]
```

```
gap> B-4;
```

```
[ 0, 1, 2, 3 ]
```

```
gap> 5*A;
```

```
[ 5, 10, 15, 20, 25 ]
```

```
gap> B/5;
```

```
[ 4/5, 1, 6/5, 7/5 ]
```

On some functions in GAP: List in GAP (8)

```
gap> W:=[1..20];  
[ 1 .. 20 ]  
gap> V:=[19..10000];  
[ 19 .. 10000 ]  
gap> Size(W);  
20  
gap> Size(V);  
9982  
gap> 1 in W;  
true  
gap> 20 in W;  
true  
gap> 25 in W;  
false  
gap> 25 in V;  
true
```

More examples of functions in GAP

```
gap> A:=[1,2,3,4,5];
```

```
[ 1, 2, 3, 4, 5 ]
```

```
gap> Sum(A);
```

```
15
```

```
gap> Product(A);
```

```
120
```

```
gap> B:=[ [1,2], [3,4] ];
```

```
[ [ 1, 2 ], [ 3, 4 ] ]
```

```
gap> Determinant(B);
```

```
-2
```

```
gap> Inverse(B);
```

```
[ [ -2, 1 ], [ 3/2, -1/2 ] ]
```

```
gap> Trace(B);
```

```
5
```

```
gap> Rank(B);
```

```
2
```


Possible error messages and how to fix them

1. What is wrong in the definition of A_5 below? How can we correct it?

```
gap> a_5:=Alternatinggroup(5);  
Error, Variable: 'Alternatinggroup' must have a value  
not in any function at *stdin*:57
```

2. What is wrong in the definition of the matrix BB below? Correct it!

```
gap> BB=[[1,2],[3,4]];  
Error, Variable: 'BB' must have a value  
not in any function at *stdin*:70
```

3. What gave the error message below? What is the right request?

```
gap> CC:=[1,2,3,4,5];;  
gap> Add(CC,6);  
gap> cc;  
Error, Variable: 'cc' must have a value  
not in any function at *stdin*:102
```

Checking for error messages in GAP

4. A student in a GAP class wants to make a list of integers from 2 up to 2000. See his GAP trials below. What should he write?

```
gap> D:=[2,...,2000];  
Syntax error: expression expected  
D:=[2,...,2000];  
^^^
```

```
gap> D:=[2,..,2000];  
Syntax error: expression expected  
D:=[2,..,2000];  
^^
```

```
gap> D:=[2...2000];  
Syntax error: Only two dots in a range  
D:=[2...2000];  
^^^
```

5. Amin wants to list all prime numbers that lie from 4050 to 4801. Which of (A) to (C) below will do the task? What should be written?

- (A) gap> Filtered([4050..4800], IsPrime);
- (B) gap> Filtered([4050..4801], Isprime);
- (C) gap> Filtered([4051..4801], Isprime);

Exercise session I

14:45 – 15:00 on August 8, 2023

Exercise 1

Open a new file in GAP called "Exercise1.g". In this new file, attempt the questions below.

1. Obtain the average of all the integers from 1 to 1000, and display your result in decimal.
2. Find the determinant, inverse and rank of the matrix A given below:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

3. Let $P = \{1, 2, \dots, 10002\}$ and $Q = \{\frac{1}{7}, \frac{2}{7}, \dots, \frac{10002}{7}\}$. What is $|P \cap Q|$?
4. How many prime numbers are there from 2 up to 999999?

5. Evaluate $\sum_{k=2}^{10} \binom{13}{k}$.

6. Assume that $10,000m^2 = 1$ hectare of land. Write a program that asks the user for the length and width (in meters) of his/her family land and gives the area of the land in hectares.
7. Write a program that ask the user for a positive integer $n > 1$ and returns both the sum of the first n positive integers and the average of the first n positive integers.

Introduction to GAP (Part II)

14:00 – 14:40 on August 9, 2023

On Groups in GAP

```
gap> G:=CyclicGroup(5);
<pc group of size 5 with 1 generators>
gap> Size(G);
5
gap> AllSubgroups(G);
[ Group([ <identity> of ... ]), <pc group of size 5
with 1 generators> ]
gap> NormalSubgroups(G);
[ <pc group of size 5 with 1 generators>, Group([ ]) ]
gap> IsAbelian(G);
true
gap> IsCyclic(G);
true
gap> IsSimple(G);
true
gap> ConjugacyClasses(G);
[ <identity> of ...^G, f1^G, f1^2^G, f1^3^G, f1^4^G ]
gap> Size(Center(G))=Size(ConjugacyClasses(G));
true
```

More on subgroups of a group

```
gap> G:=DihedralGroup(10);; gens:=GeneratorsOfGroup(G);  
[ f1, f2 ]
```

```
gap> allsub:=AllSubgroups(G);; Size(allsub);  
8
```

```
gap> G=allsub[8];
```

```
true
```

```
gap> G=allsub[1];
```

```
false
```

```
gap> H1:=Subgroup(G, [gens[1]]);
```

```
Group([ f1 ])
```

```
gap> Size(H1);
```

```
2
```

```
gap> H2:=Subgroup(G, [gens[2]]);
```

```
Group([ f2 ])
```

```
gap> Size(H2);
```

```
5
```

Normal subgroups and Factor/Quotient groups

```
gap> G:=DihedralGroup(10);; gens:=GeneratorsOfGroup(G);  
[f1, f2]
```

```
gap> ns :=NormalSubgroups(G);;  
gap> Size(ns);  
3
```

```
gap> List(ns[2]);  
[< identity > of ..., f2, f22, f23, f24]
```

```
gap> Size(ns[2]);  
5
```

```
gap> Index(G, ns[2]);  
2
```

```
gap> facgp:=FactorGroup(G, ns[2]);; Size(facgp);  
2
```


Small group library

GAP stores each group of any order uniquely in a library.

```
gap> A:=AllSmallGroups(6);  
[ <pc group of size 6 with 2 generators>,  
<pc group of size 6 with 2 generators> ]  
gap> StructureDescription(A[1]);  
"S3"  
gap> StructureDescription(A[2]);  
"C6"  
gap> B:=AllSmallGroups(6,IsAbelian,true);  
[ <pc group of size 6 with 2 generators> ]  
gap> StructureDescription(B[1]);  
"C6"  
gap> AllSmallGroups(6,IsAbelian,true,IsCyclic,false);  
[ ]  
gap> Size(AllSmallGroups(8,IsCyclic,false));  
4
```

More insight on Small group library

```
gap> G1:=SmallGroup(6,1);  
<pc group of size 6 with 2 generators>  
gap> StructureDescription(G1);  
"S3"  
gap> G2:=DihedralGroup(8);  
<pc group of size 8 with 3 generators>  
gap> StructureDescription(G2);  
"D8"  
gap> IdGroup(G2);  
[ 8, 3 ]  
gap> G3:=Group((1,2,3),(1,2));  
Group([ (1,2,3), (1,2) ])  
gap> StructureDescription(G3);  
"S3"  
gap> IdGroup(G3);  
[ 6, 1 ]
```

Testing whether two groups are isomorphic

Suppose we have two finite groups G_1 and G_2 , and want to check whether G_1 is isomorphic to G_2 . In GAP, we can do this in at least two ways: the first way is to check whether the GAP IDs of the two groups are the same while the second way is to use the function "IsomorphismGroups(G_1 , G_2)".

```
gap> G1:=CyclicGroup(2);;
gap> G2:=DirectProduct(G1,G1);;
gap> G3:=SymmetricGroup(2);;
gap> IsomorphismGroups(G1,G2);
fail
gap> IsomorphismGroups(G1,G3);
[ f1 ] -> [ (1,2) ]
gap> IsomorphismGroups(G2,G3);
fail
gap> [IdGroup(G1),IdGroup(G2),IdGroup(G3)];
[ [ 2, 1 ], [ 4, 2 ], [ 2, 1 ] ]
```

Obeying Cayley's theorem in GAP

Cayley's theorem states that every group G is isomorphic to a subgroup of the symmetric group acting on G . We shall use an example below to show how GAP obeys Cayley's theorem.

```
gap> G:=CyclicGroup(11);;
```

GAP re-writes the group G as a subgroup H of S_{11} .

```
gap> H:=Image(IsomorphismPermGroup(G)); #G re-written.  
Group([ (1,2,3,4,5,6,7,8,9,10,11) ])  
gap> A:=SymmetricGroup(11);; #This is S11.
```

Confirming Cayley's theorem on A and H .

```
gap> IsSubgroup(A,H);  
true
```

Finally, we confirm that G is isomorphic to H .

```
gap> [Size(H),IsCyclic(H)]; #Can also use IdGroup(H).  
[ 11, true ]
```

Rewriting a group as a permutation group: Thanks to Cayley's theorem!

Here, we shall rewrite each group of size 8 as a permutation group.

```
gap> DihedralGroup(IsPermGroup,8);  
Group([ (1,2,3,4), (2,4) ])
```

```
gap> QuaternionGroup(IsPermGroup,8);  
Group([ (1,5,3,7)(2,8,4,6), (1,2,3,4)(5,6,7,8) ])
```

```
gap> CyclicGroup(IsPermGroup,8);  
Group([ (1,2,3,4,5,6,7,8) ])
```

Exercise: Define $C_2 \times C_4$ and $C_2 \times C_2 \times C_2$ as permutation groups using GAP

Observing the number of generators in some cyclic groups in GAP

```
gap> G7:=CyclicGroup(3);
<pc group of size 3 with 1 generators>
gap> List(G7);
[ <identity> of ..., f1, f1^2 ]
gap> G8:=CyclicGroup(4);
<pc group of size 4 with 2 generators>
gap> List(G8);
[ <identity> of ..., f2, f1, f1*f2 ]
gap> G9:=CyclicGroup(5);
<pc group of size 5 with 1 generators>
gap> G10:=CyclicGroup(6);
<pc group of size 6 with 2 generators>
gap> G11:=CyclicGroup(8);
<pc group of size 8 with 3 generators>
```

Observing the number of generators in some dihedral groups in GAP

```
gap> G1:=DihedralGroup(6);
<pc group of size 6 with 2 generators>
gap> List(G1);
[ <identity> of ..., f2, f2^2, f1, f1*f2, f1*f2^2 ]
gap> G2:=DihedralGroup(8);
<pc group of size 8 with 3 generators>
gap> G3:=DihedralGroup(10);
<pc group of size 10 with 2 generators>
gap> G4:=DihedralGroup(12);
<pc group of size 12 with 3 generators>
gap> G5:=DihedralGroup(14);
<pc group of size 14 with 2 generators>
gap> G6:=DihedralGroup(16);
<pc group of size 16 with 4 generators>
```

Constructing a group in GAP (1)

- ▶ We could construct any group we know its presentation.
- ▶ From the fundamental theorem of presentation of groups, we know that finite groups have finite presentations. So we can easily construct finite groups in GAP.
- ▶ Some examples are displayed below for cyclic and dihedral groups of size 40.

```
gap> F:=FreeGroup("x"); #Can write F:=FreeGroup(1);  
<free group on the generators [ x ]>  
gap> x:=F.1;;  
gap> G:=F/[x^40];;  
gap> IsCyclic(G);  
true  
gap> IdGroup(G);  
[ 40, 2 ]  
gap> StructureDescription(G);  
"C40"
```


Constructing a group in GAP (2)

Recall: $D_{2n} = \langle x, y \mid x^n = 1 = y^2, (xy)^2 = 1 \rangle$ is the finite dihedral group of order $2n$.

We shall now construct the dihedral group of order 40.

```
gap> F:=FreeGroup("x","y");#Can write F:=FreeGroup(2);
<free group on the generators [ x, y ]>
gap> x:=F.1;; y:=F.2;;
gap> G:=F/[x^20,y^2,(x*y)^2];;

gap> IsDihedralGroup(G);
true
gap> Size(G);
40
gap> IdGroup(G);
[ 40, 6 ]
gap> StructureDescription(G);
"D40"
```

Defining a finitely presented group with its minimal number of generators

- ▶ As we have seen from previous slides, GAP does not always give the description of a group it stores as a function in terms of its minimal number of generators.
- ▶ **To define a group with its minimal number of generators (say k) in GAP, we construct such group as a quotient of a free group of rank k .**
- ▶ Another way we might obtain such without necessarily defining a free group is to use the command `"SimplifiedFpGroup(Image(IsomorphismFpGroup(G)));" to obtain the same group G in terms of a possible reduced number of generators.`

Description of the IsomorphismFpGroup function

IsomorphismFpGroup(G) function

- ▶ It returns an isomorphism from the given finite group G to a finitely presented group isomorphic to G . The function first chooses a set of generators of G and then computes a presentation in terms of these generators.
- ▶ Example: An illustration is given with S_5 below.

```
gap> G:=Group((1,2),(2,3,4,5));;#This is S_5.  
gap> iso:= IsomorphismFpGroup(G);  
[(4,5), (1,2,3,4,5), (1,3,2,4,5)] -> [F1, F2, F3]  
gap> fp:= Image(iso);  
<fp group on the generators [ F1, F2, F3 ]>
```

Description of the SimplifiedFpGroup function

`SimplifiedFpGroup(G)` function

- ▶ It applies Tietze transformations to a copy of the presentation of the given finitely presented group G in order to reduce it with respect to the number of generators, the number of relators, and the relator lengths.
- ▶ `SimplifiedFpGroup` returns a group isomorphic to the given one with a presentation which has been tried to simplify via Tietze transformations.

Example yielding the minimal number of generators

```
gap> G:=CyclicGroup(12);
<pc group of size 12 with 3 generators>
gap> I:=IsomorphismFpGroup(G);
[ f1, f2, f3 ] -> [ F1, F2, F3 ]
gap> J:=Image(I);
<fp group of size 12 on the generators [ F1, F2, F3 ]>
gap> K:=SimplifiedFpGroup(J);
<fp group on the generators [ F1 ]>
gap> Size(K);
12
gap> IsCyclic(K);
true
gap> GeneratorsOfGroup(K);
[ F1 ]
gap> RelatorsOfFpGroup(K);
[ F1^12 ]
```

Example yielding the minimal number of generators

```
gap> G:=DihedralGroup(12);  
<pc group of size 12 with 3 generators>  
gap> I:=IsomorphismFpGroup(G);  
[ f1, f2, f3 ] -> [ F1, F2, F3 ]  
gap> J:=Image(I);  
<fp group of size 12 on the generators [ F1, F2, F3 ]>  
gap> K:=SimplifiedFpGroup(J);  
<fp group on the generators [ F1, F2 ]>  
gap> Size(K);  
12  
gap> IsDihedralGroup(K);  
true  
gap> GeneratorsOfGroup(K);  
[ F1, F2 ]  
gap> RelatorsOfFpGroup(K);  
[ F1^2, (F1*F2)^2, F2^6 ]
```

What is GAP doing to obtain such minimal number of generators?

```
gap> G:=DihedralGroup(12);  
<pc group of size 12 with 3 generators>  
gap> H:=Image(IsomorphismFpGroup(G));  
<fp group of size 12 on the generators [ F1, F2, F3 ]>  
gap> IsDihedralGroup(H);  
true
```

What is actually happening in GAP?

```
gap> IsomorphismSimplifiedFpGroup(H);  
[ F1, F2, F3 ] -> [ F1, F2, F2^2 ]
```

To simply the above operation well, we get:

```
gap> S:=SimplifiedFpGroup(H);  
<fp group on the generators [ F1, F2 ]>  
gap> IsDihedralGroup(S);  
true
```

We can define more groups in GAP

```
gap> G1:=Group((1,2),(2,3,4,5));
Group([ (1,2), (2,3,4,5) ])
gap> Size(G1);
120
gap> IdGroup(G1);
[ 120, 34 ]
gap> IsSymmetricGroup(G1);
true
gap> G2:=Group((1,2,3),(2,3,4));
Group([ (1,2,3), (2,3,4) ])
gap> Size(G2);
12
gap> [IsSymmetricGroup(G2),IsAlternatingGroup(G2)];
[ false, true ]
gap> IdGroup(G2);
[ 12, 3 ]
```


A little about rings

There are at least two ways of defining Z_5 in GAP.

```
gap> R1:=Integers mod 5;; R2:=GF(5);; R1=R2;  
true
```

```
gap> Elements(R1);  
[0 * Z(5), Z(5)0, Z(5), Z(5)2, Z(5)3]
```

To see what integer $Z(5)$ represents (mod 5) type:

```
gap> Int(Z(5));  
2
```

We now create the indeterminate over the ring R1 as follows:

```
gap> x:= X(R1, "x");  
x
```

We can now set up a polynomial ring:

```
gap> P1:=PolynomialRing(R1);  
GF(5)[x]
```

```
gap> IsIrreducible(x2 - 4);  
false
```

```
gap> Factors(x2 - 4);  
[x + Z(5), x + Z(5)3]
```

A little about vector spaces

```
gap> intg:=Integers;; Size(intg);  
infinity
```

```
gap> f25:= GF(25);; Size(f25);  
25
```

```
gap> F:= Rationals;;
```

```
gap> V:= VectorSpace( F, [[1, 1, 1], [0, 1, 1]] );;
```

```
gap> [2, 3, 2] in V;
```

```
false
```

```
gap> [1, 2, 2] in V;
```

```
true
```

```
gap> B:=Basis(V);;
```

```
gap> Size(Basis(V));
```

```
2
```

```
gap> B[1];
```

```
[1, 1, 1]
```

```
gap> B[2];
```

```
[0, 1, 1]
```

Exercise session II

14.40 – 15:00 on August 9, 2023

Exercise II

1. Give the GAP ID and description of the following groups:

(i) $G_1 = \langle x, y \mid x^6 = 1 = y^2, xy = yx^{-1} \rangle$;

(ii) $G_2 = \langle x, y \mid x^6 = 1, x^3 = y^2, xy = yx^{-1} \rangle$;

(iii) $G_3 = \langle x, y \mid x^3 = 1 = y^4, xyx^{-1}y^{-1} = 1 \rangle$.

2. Identify the finite group whose GAP ID is [32, 51]. Then write the group as a permutation group.

3. Give a presentation of the group whose GAP ID is [32, 16].

4. Find the size of the minimal generating set, GAP ID and Structure description of the following groups:

(i) $G_1 = \langle (1, 2, 3), (4, 5, 6), (5, 6, 7), (8, 9, 10) \rangle$;

(ii) $G_2 = \langle (1, 2, 3), (1, 2), (4, 5, 6), (5, 6, 7), (8, 9, 10) \rangle$.

5. What is the ratio of the number of abelian groups of order 100 to the number of nonabelian groups of order 100?

Introduction to GAP (Part III)

08:00 – 09:00 on August 10, 2023

Regression tests

The regression tests we do in GAP include the following.

- ▶ **if** bool-expr1 **then**
statements1
elif bool-expr2 **then**
statements2
else
statements3
fi;
- ▶ **for** simple-var in list-expr **do**
statements
od;
- ▶ **while** bool-expr **do**
statements
od;
- ▶ **repeat** statements **until** bool-expr;

1. Obtain a list of all the odd numbers from 1 up to 1000

1. Obtain a list of all the odd numbers from 1 up to 1000

Method I (Writing the code from scratch)

```
L:= [1..1000];; Od:= [];;  
for i in L do  
  if i mod 2=1 then  
    Add(Od,i);  
  fi;  
od;  
Print(Od); #This will give the required list.
```

Method IIa (Using an inbuilt function in GAP)

```
L:= [1..1000];; Od:= [];;  
for i in L do  
  if IsOddInt(i) then  
    Add(Od,i);  
  fi;  
od;  
Print(Od); #This will give the required list.
```

Method IIb (Using an inbuilt function in GAP)

```
Filtered([1..1000], IsOddInt);
```


2. Write a program that will return a list of GAP IDs of all the nonabelian groups of order $2p^2$ for any odd prime p the user chooses.

```
Ord:=function(p)
local GG, A, G;
GG:=AllSmallGroups(2*p^2,IsAbelian,false);
A:=[]; for G in GG do Add(A,IdGroup(G)); od;
return A; end;
```

```
gap> Ord(3);
[ [ 18, 1 ], [ 18, 3 ], [ 18, 4 ] ]
gap> Ord(5);
[ [ 50, 1 ], [ 50, 3 ], [ 50, 4 ] ]
```

Exercise

(a) Suggest another program for the above question.

(b) Make a conjecture on the number of nonabelian groups of order $2p^2$ for any odd prime p . Does your conjecture also hold for all nonabelian groups of order $2p^n$, where $n \geq 3$?

3. Write a program that takes as an input, scores S_1, S_2, S_3, S_4 and S_5 gotten by a Masters student in coursework and determines the student's result as Distinction, Merit, Pass or Fail. The student gets:

Distinction if the average score is at least 75%;

Merit if the average score is less than 75% but at least 65%;

Pass if the average score is less than 65% but at least 50%;

Fail if otherwise.

3. Write a program that takes as an input, scores S_1, S_2, S_3, S_4 and S_5 gotten by a Masters student in coursework and determines the student's result as Distinction, Merit, Pass or Fail. The student gets:

Distinction if the average score is at least 75%;

Merit if the average score is less than 75% but at least 65%;

Pass if the average score is less than 65% but at least 50%;

Fail if otherwise.

```
ResSol:=function(S1, S2, S3, S4, S5)
local Av;
Av:=Average([S1,S2,S3,S4,S5]);
if Av>=75 then Print("DISTINCTION");
elif Av>=65 then Print("MERIT");
elif Av>=50 then Print("PASS");
else Print("FAIL"); fi;
end;
```

4. Define a function that will take in any finite group from the user and returns all subgroups of index 3 if the group has such and returns "[]" if the group has no subgroup of index 3.

4. Define a function that will take in any finite group from the user and returns all subgroups of index 3 if the group has such and returns "[]" if the group has no subgroup of index 3.

```
Ind3:=function(G)
local HH,H,AA;
if (Size(G) mod 3)>0 then return [];
else HH:=AllSubgroups(G); AA:=[];
for H in HH do if Index(G,H)=3 then Add(AA,H); fi; od;
return AA; fi;
end;
```

```
gap> Ind3(SymmetricGroup(3));
[ Group([ (2,3) ]), Group([ (1,2) ]), Group([ (1,3) ]) ]
gap> Ind3(CyclicGroup(8));
[ ]
gap> F:=FreeGroup(1);; x:=F.1;; G:=F/[x^24];; Ind3(G);
[ Group([ f1^3 ]) ]
```

5. Two students (Student A and Student B) were asked to write a program that will take as an input a finite group G and an element g of G , and return the order of g . Below are their attempts.

Student A

```
OrdAGg:=function(G,g)
local n;
n:=1;
while (g^n=Identity(G))=false do n:= n+1; od;
return n; end;
```

Student B

```
OrdBGg:=function(G,g)
local n, a;
n:= 1; a:=g;
while (a=Identity(G))=false do n:= n+1; a:=a*g; od;
return n; end;
```

- (a) Would any of the above two programs will yield a faster result when $|G|$ is large enough?
- (b) Is there an easier way of answering this question without defining a function in GAP?

Class Exercise

1. Define a function that will take inputs as two sets A and B and return the cardinality of the union of A and B (Please do not use any function like `Union`, `UniteSet`, `UnionSet` et-cetera).
2. Write a program that asks the user for any finite group G and a positive integer $k > 1$ and returns both a list of all the elements of order k in G and the number of elements of order k in G .
3. Define a function that will take inputs as integers a, b, c and prints "Your discriminant is negative!" if $b^2 - 4ac < 0$, but returns the value $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ if $b^2 - 4ac \geq 0$.
4. Define a function that will take as an input a finite group G and return whether or not the group is simple. [Please do not use the GAP command "IsSimple(G)" in your solution.]
5. Write a program that reads an n digit number and returns the sum of the digits of the number.

Class Exercise continues

6. Write a program that takes as an input a positive integer n , and returns the ratio of the nonabelian groups of order n to the abelian groups of order n . [Test your program for $n = 1, 5, 6, 25$ and 100 .]

7. Write a program that takes as an input scores S_1, S_2, S_3, S_4 and S_5 gotten by a Masters student in coursework and determines the student's result as Distinction, Merit, Pass or Fail under the condition given below. Writing Avr for the average of the five scores, the students gets:

Distinction if $Avr \geq 75\%$, and $S_i \geq 65\% \forall i \in \{1, 2, 3, 4, 5\}$;

Merit if $65\% \leq Avr < 75\%$, and $S_i \geq 50\% \forall i \in \{1, 2, 3, 4, 5\}$;

Pass if $50\% \leq Avr < 65\%$, and $S_i \geq 40\% \forall i \in \{1, 2, 3, 4, 5\}$;

Fail if otherwise.

8. Write a program that will ask the user for his/her ATM pin. If the user enters the ATM pin wrongly three times, it will type "You entered a wrong pin three times, so we blocked your card. Please see customer care". If the user enters the ATM pin correctly, it will print "Your ATM pin is correct; so proceed.". [Hint: Use 1701 as your default ATM pin. Also, use the function "InputFromUser" somewhere in your program.]

Exercise session III

09.00 – 09:25 on August 10, 2023

Exercise III

1. Write a program that takes as an input a finite group G , and returns the number of conjugacy classes in G , the center of G and the derived subgroup of G .
2. Write a program that takes as an input a prime number p and a natural number n , and returns the elementary abelian group p -group of size p^n . [Do not use the elementary abelian group function.]
3. Write a program that takes as an input a finite group G , and returns a presentation of G in the form $[\text{gen}, \text{rel}]$, where gen is the set of generators of the group G and rel is the set of relators of G .
4. Write a program that takes as an input two finite groups G and H , and returns true if G is isomorphic to H and false if G and H are not isomorphic.
- 5a. How many groups are there of order less than or equal to 1000?
 - b. How many nonabelian simple groups are there of order up to 1000?
 - c. How many simple groups of order less than or equal to 1000 are there?
 - d. How many nonabelian groups of order 1024 are there?

Exercise III continues

6a. Identify the groups given by the following presentations:

$$G_1 = \langle x, y, z : x^3 = y^2 = z^2 = (xz)^2 = (xy)^3 = (yz)^3 = 1 \rangle;$$

$$G_2 = \langle x, y, z : x^3 = y^2 = z^2 = (xz)^2 = (xy)^3 = (yz)^2 = 1 \rangle;$$

$$G_3 = \langle x, y, z : x^3 = y^2 = z^2 = (xz)^2 = (xy)^2 = (yz)^2 = 1 \rangle;$$

$$G_4 = \langle x, y, z : x^3 = y^3 = z^3 = (xz)^2 = (xy)^2 = (yz)^2 = 1 \rangle;$$

$$G_5 = \langle x, y, z : x^3 = y^3 = z^2 = (xz)^2 = (xy)^2 = (yz)^2 = 1 \rangle;$$

$$G_6 = \langle x, y, z : x^4 = y^3 = z^2 = (xz)^2 = (xy)^2 = (yz)^2 = 1 \rangle;$$

$$G_7 = \langle x, y, z : x^5 = y^3 = z^2 = (xz)^2 = (xy)^2 = (yz)^2 = 1 \rangle.$$

b. Give a presentation of each of the groups in 6a in terms of the minimal number of generators, if not written already.

7. List finite groups S and G of the same size, such that S and G have the same number of elements of order n for any n dividing $|S|$, but G is not isomorphic to S . Obtain all such G and S for $|G| = |S| \leq 100$.

8. Stefan has a large pile of socks that he wants his son to pair by colour after washing. Write a program in GAP, that given a list of integers representing the colour of each sock, determines how many pairs of socks with matching colours there are. For instance, the list $[1, 2, 3, 4, 4, 4, 5, 9]$ will return 1 pair of socks, the list $[3, 3, 1, 2, 3, 4, 1, 1, 3, 5, 5]$ will return 4 and $[2, 3, 4, 5]$ will return 0.

Project activities

09:25–09:55 on August 10, 2023

Project: Attempt Conjecture 1 or 2 below, but not both

Project Activities: 30 minutes

1. Attempt the assigned conjecture for your group (20 minutes).
2. We would need two volunteers, one from each group, to tell us the steps s/he took in tackling the conjecture and what s/he thinks about the validity of the conjecture. (10 minutes: 5 minutes each).

Conjecture 1 (B. Curtin and G. R. Pourgholi)

Let G and H be finite groups of the same order with

$$\sum_{g \in G} \varphi(|g|) = \sum_{h \in H} \varphi(|h|).$$

Suppose that G is simple. Is H necessarily simple?

(Ref: Question 19.25 of the 19th issue (2018) of the Kourovka Notebook.)

Note. $\varphi(n)$ is the number of positive integers up to n which are coprime to n .

Conjecture 2 (L. J. Taghvasani and M. Zarrin)

Let G be a finite nonabelian simple group. Then $|G(p)| \neq |G(q)|$ for all distinct prime divisors p and q of $|G|$, where $G(p) := \{g \in G : o(g) = p\}$.

(Ref: Monatshefte für Mathematik 182(3) (2017), 731–736.)

Concluding remarks

09:55–10:00 on August 10, 2023

Concluding remarks

- ▶ Do not forget the help function in GAP.
- ▶ If you need further assistance, you can use any of the two platforms below to ask your question (remember to send your attempt too!):
 - (a) Mathematics Stack Exchange
(<https://math.stackexchange.com/tags/gap/info>);
 - (b) Send an email to forum@gap-system.org.

You can also subscribe to the GAP Forum mailing list by signing up at <https://lists.uni-kl.de/gap/info/forum>. This will help you to see questions asked by others, as well as any answers and/or suggestions given by various individuals. You can also answer questions asked by others too!

Enjoy using GAP!