GAC2010 — Groups, Actions and Computations

GAP session 1

Getting to know GAP

These tasks are meant to keep **everybody** busy for at least an hour. Do **not dispair** if you finish only part of them. There are hints on this sheet, however, if you get **stuck**, then **ask** someone.

1. Find out how to start GAP, type in commands and compute $123456 \cdot 654321$.
   Hints: Type `gap` or `gap4` at the command prompt, end your GAP commands with a semicolon (`;`).
2. Find out how to use GAP's help system, look up the `Orbit` command.
   Hints: Type a question mark (`?`) followed by a word to find a manual section heading beginning with that word. Type two question marks (`??`) followed by a word to find all manual section headings containing that word.
3. Read about the `InputLogTo` command. Log your input to a file.
4. Find out whether the product of the two permutations $(1,2)$ and $(2,3)$ is $(1,2,3)$ or $(1,3,2)$. This tells you whether GAP prefers to act from the left or right.
5. Compute the factorial of 1000, that is $1000! = 1 \cdot 2 \cdots 999 \cdot 1000$.
   Hints: You can use the library function. Later we will write a function to do this on our own.
6. Try to divide by zero and see what happens.
   Hints: Read about the so-called "break loop" under `?break loop`.
7. Find out how to edit a text file on your computer, call it "`test.g`" and read it into GAP using the `Read` command.
8. It is now time to write your first function, try this one:

```
MyFirst := function(a,b)
  local c;
  c := a^2 + b^2;
  Print("It worked: ",a," and ",b," gives ",c,"\n");
  return c;
end;
```

9. Write a function computing factorials.
   Hints: Read about the `for` and `while` and `if` commands.
10. Put an `Error` statement into the above function.
    Hints: GAP enters a break loop during execution. This is very useful for debugging.
11. Compute the orbit of the pair $[3,6]$ under the action on pairs of the permutation group generated by the three permutations
    $(1,2,3,4,5,6,7,8,9,10,11)$, $(3,7,11,8)(4,10,5,6)$ and $(1,12)(2,11)(3,6)(4,8)(5,9)(7,10)$.
12. Write a program to find a solution to the "8-queens problem": How can you place 8 queens on a chess board such that none of them attacks another one (i.e. no two are in the same row, column or diagonal).
    Hints: You probably have to learn about **lists** for this. Try `?lists` for this.