Computer Algebra, week 1, lecture 2: Operations with polynomials and the use of quotes ’ ’ in Maple.

Let F be some polynomial in x, for example:

```
> restart;  # This command un-assigns all variables, as if you had
  restarted Maple.
> F:=x^5+x+1;
```

$$F := x^5 + x + 1$$

When we assigned F, the variable x had not yet been assigned. If x had been assigned, for example if x had the value 1, then F wouldn’t have been what it is now, but would have had the value 3.

```
> x:=1;
```

$$x := 1$$

```
> F;
```

$$3$$

However, we did not first assign x and then F, but first F and then x. That makes a difference, because even though F now looks like 3, it really is still x^5+x+1 which just happens to evaluate to 3 but is not quite the same thing as 3. This becomes visible if you change x. Changing x will have no impact on 3, but will have an impact on F, and thus F is not completely the same as 3.

```
> x:=12;
```

$$x := 12$$

```
> F;
```

$$248845$$

```
> x:=y;
```

$$x := y$$

```
> F;
```

$$y^5 + y + 1$$

The "restart" command wipes out the values of all variables. But what do we do if we want to wipe out the value of x (i.e. undo just the assignment of x and not undo any other assignments). So we want x to be x again, not something else:

```
> x:=x;
```

$$x := y$$

This didn’t work. Why? Well, when you do:
  x := x;
then Maple will first evaluate the right-hand side. But the right-hand side x had value y (we will say: x evaluates to y), so Maple replaces the right-hand side by y. Because of that, the command: x:=x; doesn’t do anything.
To unassign x, we have to do the following:

```
> x:=’x’;
```

$$x := x$$

```
> F;
```

$$x^5 + x + 1$$

```
> x;
```

$$x$$

In Maple, the forward quotes are used when you don’t want to evaluate an expression. Lets say that x has the value 3, and you do x:=x; Then the right-hand side will be evaluated to 3, and the assignment you are doing will be just x:=3; Now what do you do if you want x to be x again, and not 3? The only way to do that is to make sure that the right-hand side of x:=x; does not get evaluated, so that Maple will assign x to x, instead of setting x to be the value of x which was 3. That is what the quotes ’ ’ do:
these quotes cause the expression to be not evaluated.
Let’s look at some examples of what the quotes ’ ’ do in Maple.

```
> d:=3;
```

$$d := 3$$

```
> d;
```

$$3$$

Typing d; will get you the same result as typing 3; because Maple evaluates d to 3, and then displays the result. However, ’d’ will get evaluated to d. In general, whenever you type   expression; then Maple will evaluate that expression as far as it can, so d; becomes 3. But when you type ’expression’ then Maple will evaluate it to expression, and will do no further evaluation to that. Let’s see some examples of that:

```
> '''d''';  # Looks like:  'expression' where expression = ''d''
  Hence:
```

$$''d''$$

```
> %; # Now we're evaluating 'expression' where expression = 'd' so
  we get:
```

$$'d'$$

```
> %; # Each time we evaluate, we lose a pair of quotes, so we get:
```

$$d$$

```
> %; # When there are no more quotes left, then Maple will evaluate
  the expression:
```

$$3$$

```
> %; # and of course 3 evaluates to 3:
```

$$3$$

This shows the effect of the quotes. At first we had something of the form ’expression’ where expression was ’’d’’, so that will be the output. When you have Maple evaluate the output ’’d’’, using the %; then the evaluation will remove another pair of quotes. When there are no quotes, Maple will evaluate as far as it can. Lets see an example of that:

```
> a:=b;
```

$$a := b$$

```
> b:=c;
```

$$b := c$$

```
> c:='d';
```

$$c := d$$

```
> a;
```

$$3$$

Explanation: when you ask for a, Maple evaluates it as far as it can. And a evaluates to b, but then Maple thinks: hey, I can evaluate that further, and there are no quotes that tell me not to evaluate, so I evaluate further. It evaluates the b into a c, and evaluates that into d (if I hadn't put quotes around the d, then the value of c would not have been d, it would have been 3, so in that case when it evaluates c the result wouldn't have been d, it would have been 3). But the d that Maple ends up with has no quotes around it (those were removed in the evaluation during the command c:='d';) so Maple will keep on evaluating, and then d evaluates to 3. So a evaluates to 3.

```
> c:=''d'';
```

$$c := 'd'$$

```
> a;
```

$$d$$

Now a, evaluates to b, which evaluates to c, which evaluates to 'd', and that evaluates do d and Maple will stop there because the quotes tell Maple to stop.

```
> %;
```

$$3$$

At this point, the variables F, a, b, c, d have been assigned. And x has been assigned, but then later unassigned. Suppose I want to clear the value of all variables, I want to unassign all of them. I could do this as follows:

```
> a:='a';
```

$$a := a$$

```
> b:='b';
```

$$b := b$$

```
> c:='c';
```

$$c := c$$

```
> d:='d';
```

$$d := d$$

```
> F:='F';
```

$$F := F$$

Or I could do it shorter by: a,b,c,d,F := 'a','b','c','d','F';
A quicker way to unassign all variables is the following command:

```
> restart;
```

Lets look at some operations with polynomials.

```
> F:=(x^2+x+1)^5 * (x-1)^2*(x-2)*(x-2) * (x-3)^3;
```

$$F := (x^2+x+1)^5\,(x-1)^2\,(x-2)^2\,(x-3)^3$$

Maple acts lazy here, it pretty much just returns what I typed and does almost no computation. The only thing it did was to replace (x-2)*(x-2) by (x-2)^2. You may expect it would multiply out all these products, but it didn't. To expand a product, you have to tell Maple that that's what you want, otherwise it will think that the form you gave is the polynomial is the form you want to see.

```
> F:=expand(F);
```

$$F := -108 - 171\,x^2 + 242\,x^4 + 310\,x^3 - 388\,x^6 + 481\,x^5 - 490\,x^8 - 140\,x^7 - 24\,x^{10} + 312\,x^9 - 108\,x - 160\,x^{12} + 222\,x^{11} - 43\,x^{14} + 40\,x^{13} + 34\,x^{15} + x^{17} - 10\,x^{16}$$

In this particular example you can see that there was good reason not to expand, because the expanded form is longer than the original form. What if you only have the expanded form, and you want the factored form?

```
> factor(F);
```

$$(x^2 + x + 1)^5\,(x-1)^2\,(x-2)^2\,(x-3)^3$$

So Maple can expand polynomials and factor polynomials. Both go quickly. The factored form is often shorter, but not always, see:

```
> f:=x^100-1;
```

$$f := x^{100} - 1$$

```
> factor(f);
```

$$(x-1)\,(x^4 + x^3 + x^2 + x + 1)\,(x^{20} + x^{15} + x^{10} + x^5 + 1)\,(x+1)\,(1 - x + x^2 - x^3 + x^4)\,(1 - x^5 + x^{10} - x^{15} + x^{20})\,(1 + x^2)\,(x^8 - x^6 + x^4 - x^2 + 1)\,(x^{40} - x^{30} + x^{20} - x^{10} + 1)$$

```
> expand(%);
```

$$x^{100} - 1$$

The polynomials f and F have a factor in common. The greatest common divisor can be computed by the gcd.

```
> gcd(F,f);
```

$$x - 1$$

Whenever a polynomial F has a root alpha of multiplicity e, with e>0, then the derivative F' has a root alpha with multiplicity e-1. Now F has roots with multiplicities 5, 2 and 3, so the derivative F' will also have those roots, but with multiplicities 4, 1 and 2. It can also have other roots that F does not have (remember that the roots of F' are the critical points of F).

```
> F;
```

$$-108 - 171\,x^2 + 242\,x^4 + 310\,x^3 - 388\,x^6 + 481\,x^5 - 490\,x^8 - 140\,x^7 - 24\,x^{10} + 312\,x^9 - 108\,x - 160\,x^{12} + 222\,x^{11} - 43\,x^{14} + 40\,x^{13} + 34\,x^{15} + x^{17} - 10\,x^{16}$$

```
> diff(F,x);
```

$$-342\,x + 968\,x^3 + 930\,x^2 - 2328\,x^5 + 2405\,x^4 - 3920\,x^7 - 980\,x^6 - 240\,x^9 + 2808\,x^8 - 108 - 1920\,x^{11} + 2442\,x^{10} - 602\,x^{13} + 520\,x^{12} + 510\,x^{14} + 17\,x^{16} - 160\,x^{15}$$

```
> gcd(% , %%);
```

$$x^{12} - 5\,x^{11} + 3\,x^{10} + 3\,x^9 + 27\,x^8 - 9\,x^7 - 27\,x^6 - 75\,x^5 - 27\,x^4 + 5\,x^3 + 53\,x^2 + 33\,x + 18$$

```
> factor(%);
```

$$(x-1)\,(x-2)\,(x-3)^2\,(x^2 + x + 1)^4$$

As you can see, when we took gcd(F, diff(F,x)) then the multiplicity 5 of the factor x^2+x+1 became multiplicity 4, multiplicity 3 for the factor x-3 became multiplicity 2, and the roots 1 and 2 (factors x-1 and x-2) with multiplicity 2 now have multiplicity 1.

With the gcd in factored form, we can easily see that it is correct. We can differentiate twice, or 3 times, or more, and see what happens then:

```
> G1 := F;
```

$$G1 := -108 - 171\,x^2 + 242\,x^4 + 310\,x^3 - 388\,x^6 + 481\,x^5 - 490\,x^8 - 140\,x^7 - 24\,x^{10} + 312\,x^9 - 108\,x - 160\,x^{12} + 222\,x^{11} - 43\,x^{14} + 40\,x^{13} + 34\,x^{15} + x^{17} - 10\,x^{16}$$

```
> factor(%);
```

$$(x-1)^2\,(x-2)^2\,(x-3)^3\,(x^2+x+1)^5$$

```
> G2 := gcd( G1 , diff(G1,x) );
```

$$G2 := x^{12} - 5\,x^{11} + 3\,x^{10} + 3\,x^9 + 27\,x^8 - 9\,x^7 - 27\,x^6 - 75\,x^5 - 27\,x^4 + 5\,x^3 + 53\,x^2 + 33\,x + 18$$

```
> factor(%);
```

$$(x-1)\,(x-2)\,(x-3)^2\,(x^2+x+1)^4$$

```
> G3 := gcd( G2, diff(G2,x) );
```

$$G3 := x^7 - 3\,x^5 - 11\,x^4 - 15\,x^3 - 15\,x^2 - 8\,x - 3$$

```
> factor(%);
```

$$(x-3)\,(x^2+x+1)^3$$

```
> G4 := gcd(G3, diff(G3,x) );
```

$$G4 := x^4 + 2\,x^3 + 3\,x^2 + 2\,x + 1$$

```
> factor(%);
```

$$(x^2+x+1)^2$$

```
> G5 := gcd(G4, diff(G4,x));
```

$$G5 := x^2 + x + 1$$

```
> factor(%);
```

$$x^2 + x + 1$$

```
> has(G5,x);  # Same as checking if degree(G5,x) is >0 or not.
```

true

```
> G6 := gcd(G5, diff(G5,x) );
```

$$G6 := 1$$

```
> has(G6,x);
```

false

We have seen that with differentiation and gcd's we can find factors of F that have different multiplicities. That is what's called a **square-free factorization.**

```
> F;
```

$$-108 - 171\,x^2 + 242\,x^4 + 310\,x^3 - 388\,x^6 + 481\,x^5 - 490\,x^8 - 140\,x^7 - 24\,x^{10} + 312\,x^9 - 108\,x$$

$$- 160 x^{12} + 222 x^{11} - 43 x^{14} + 40 x^{13} + 34 x^{15} + x^{17} - 10 x^{16}$$

```
> sqrfree(F);
```

$$[1, [[x^2 - 3 x + 2, 2], [x - 3, 3], [x^2 + x + 1, 5]]]$$

```
> G:=20*F;
```

$$G := -2160 - 3420 x^2 + 4840 x^4 + 6200 x^3 - 7760 x^6 + 9620 x^5 - 9800 x^8 - 2800 x^7 - 480 x^{10} + 6240 x^9 - 2160 x - 3200 x^{12} + 4440 x^{11} - 860 x^{14} + 800 x^{13} + 680 x^{15} + 20 x^{17} - 200 x^{16}$$

```
> v:=sqrfree(G);
```

$$v := [20, [[x^2 - 3 x + 2, 2], [x - 3, 3], [x^2 + x + 1, 5]]]$$

To recover a polynomial G from its square-free factorization, we can use the following command (see the previous worksheet for an explanation of the command mul).

```
> v[1] * mul(i[1]^i[2], i=v[2]);
```

$$20 (x^2 - 3 x + 2)^2 (x - 3)^3 (x^2 + x + 1)^5$$

Here v[1] = the constant factor.
And i runs through the list [[factor1, mult1], [factor2, mult2], ...] so i[1] = a factor,   and i[2] = the multiplicity of that factor.

The square-free command sqrfree found the following factors:

```
> seq(i[1], i=v[2]);
```

$$x^2 - 3 x + 2, x - 3, x^2 + x + 1$$

It could compute those factors with diff and gcd because they all had different multiplicities. However, x-1 and x-2 had the same multiplicity, and because of that, x-1 and x-2 could not be seperated by just "diff" and "gcd". So sqrfree does not factor x^2-3*x+2 into (x-1)*(x-2).

```
> G;
```

$$-2160 - 3420 x^2 + 4840 x^4 + 6200 x^3 - 7760 x^6 + 9620 x^5 - 9800 x^8 - 2800 x^7 - 480 x^{10} + 6240 x^9 - 2160 x - 3200 x^{12} + 4440 x^{11} - 860 x^{14} + 800 x^{13} + 680 x^{15} + 20 x^{17} - 200 x^{16}$$

```
> factor(G);
```

$$20 (x - 1)^2 (x - 2)^2 (x - 3)^3 (x^2 + x + 1)^5$$

```
> op(%);  # op converts this product into a sequence:
```

$$20, (x - 1)^2, (x - 2)^2, (x - 3)^3, (x^2 + x + 1)^5$$

```
> %[3];
```

$$(x - 2)^2$$

```
> op(%); # op converts this a^b into a sequence a,b
```

$$x - 2, 2$$

As you can see, you can get the individual factors from the product with the op command. That’s not always convenient, it’s sometimes easier to use the command factors which does that for you.

```
> v:=factors(G);
```

$$v := [20, [[x - 3, 3], [x^2 + x + 1, 5], [x - 2, 2], [x - 1, 2]]]$$

```
> v[1];
```

$$20$$

```
> v[2];
```

$$[[x-3, 3], [x^2+x+1, 5], [x-2, 2], [x-1, 2]]$$

Now v is a list with 2 elements. v[1] is the constant factor (which is not factored by the factors command), and v[2] is the list of factors with their multiplicities.

The commands **factor** and **factors** factor polynomials with coefficients that are rational numbers. Since 20 is a unit in the rational numbers, it will not be factored by those commands. To factor integers there is a different command:

```
> ifactor(20);
```

$$(2)^2\ (5)$$

```
> igcd(20,50);
```

$$10$$

```
> igcd(10^100-1, 1000!);
```

$$27885821139$$

```
> ifactor(%);
```

$$(3)^2\ (11)\ (41)\ (101)\ (251)\ (271)$$

```
>
```