

Pr.13/17.2

```
[ > z1 := 5 + 12*I; # Resp. z1 := 5 + 12I
[ > z2 := 3 - 7*I; # Resp. z2 := 3 - 7I
[ > z1*conjugate(z2); # Resp. -69 + 71I
[ > conjugate(z1)*z2; # Resp. -69 - 71I
```

By taking the conjugate of the first product you get the second.

```
[ > 1/abs(z1); # Resp. 1/13
[ > evalf[3](abs(z1) + abs(z2) - abs(z1 + z2)); # Resp. 11.2
```

This must be nonnegative because of the triangle inequality.

```
[ > Re(z1^3); # Resp. -2035
[ > (Re(z1))^3; # Resp. 125
[ > Im((z1 - z2)/(z1 + z2)); # Resp. 142/89
```

Pr.13/17.4 Type `?polar` for information, in particular, on the multiplication in polar form. Use `evalc` for converting back.

```
[ > z1 := polar(1 - I); # Resp. z1 := polar(sqrt(2), -1/4*pi)
[ > evalc(%); # Resp. 1 - I
[ > z2:= polar(-3 - 3*I); # Resp. z2 := polar(3*sqrt(2), -3/4*pi)

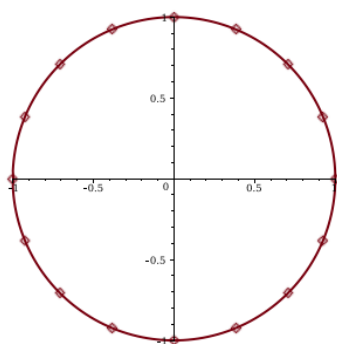
[ > z1*z2; # Resp. polar(sqrt(2), -1/4*pi) polar(3*sqrt(2), -3/4*pi)
[ > simplify(%); # Resp. -6
[ > (1 - I)*(-3 - 3*I); # Resp. -6
[ > polar(-15); # Resp. polar(15, pi)
[ > polar((1 - I)/(1 + I)); # Resp. polar(1, -1/2*pi)
[ > polar(((6 + 8*I)/(4 - 3*I))^2); # Resp. polar(4, pi)
```

Pr.13/17.6 The four solutions are obtained by typing

```
[ > solve(z^4 + (6*I)*z^2 + 3*z^2 - 8 + 6*I);
-1 + I, 1 - I, -1 + 2I, 1 - 2I,
```

Pr.13/17.8 The first four values $1, -1, i, -i$ are obvious. These are the 4th roots of unity. Together with the next four values they make up the 8th roots of unity. Hence their position on the unit circle is now clear. The last eight values lie “in the middle” between the ones just discussed.

```
[ > solve(z^16 = 1);
> S := evalf(%);
S := 1.0, -1.0, 1.0I, -1.0I, 0.7071067810 + 0.7071067810I, -0.7071067810
- 0.7071067810I, 0.7071067810 - 0.7071067810I, -0.7071067810 + 0.7071067810I,
0.9238795325 - 0.3826834323I, -0.9238795325 + 0.3826834323I, 0.9238795325
+ 0.3826834323I, -0.9238795325 - 0.3826834323I, 0.3826834323 - 0.9238795325I,
-0.3826834323 + 0.9238795325I, 0.3826834323 + 0.9238795325I,
-0.3826834323 - 0.9238795325I
> with(plots):
> complexplot([S], x=-1.5..1.5, style=point, symbolsize = 20,
scaling = constrained);
```



Pr.13/17.12 Yes.

```
[ > z := x + I*y;
> f := evalc(Re(z^3) + I*Im(z^3));
f := x^3 - 3xy^2 + I(3x^2y - y^3)
> u := evalc(Re(f)); # Resp. u := x^3 - 3xy^2
> v := evalc(Im(f)); # Resp. v := 3x^2y - y^3
> CR1 := diff(u, x) - diff(v, y); # Resp. CR1 := 0

[ > CR2 := diff(u, y) + diff(v, x); # Resp. CR2 := 0
```

Pr.13/17.20 Type

```
[ > evalc((2*I)^(2*I));          # Resp. e-π cos(2 ln(2)) + I e-π sin(2 ln(2))
[ > evalc(4^(3-I));              # Resp. 64 cos(2 ln(2)) - 64 I sin(2 ln(2))
[ > evalc((3 + 6*I)^I);
    e-arctan(2) cos( $\frac{1}{2} \ln(45)$ ) + I e-arctan(2) sin( $\frac{1}{2} \ln(45)$ )
[ > evalc(I^(3/2));              # Resp.  $-\frac{1}{2}\sqrt{2} + \frac{1}{2}I\sqrt{2}$ 
```

Pr.14.1 In the last command try without `evalc`.

```
[ > z := 1 + I + (1 + 2*I)*t;    # Resp. z := 1 + I + (1 + 2 I) t

[ > zdot := diff(z, t);          # Resp. zdot := 1 + 2 I
[ > int(evalc(Im(z)*zdot), t = 0..1); # Resp. 2 + 4 I
```

Pr.14.2 Type a representation of the path C and its derivative, then the given function on C , and finally its integral (minus sign because we are going clockwise).

```
[ > z := 1 + 3*exp(-I*t);        # Resp. z := 1 + 3 e-It
[ > zdot := diff(z, t);          # Resp. zdot := -3 I e-It
[ > f := 7/(z + I) - 5/(z + I)^2;
    f :=  $\frac{7}{2I + 3e^{-It}} - \frac{5}{(2I + 3e^{-It})^2}$ 
[ > int(evalc(f*zdot), t = 0..2*Pi); # Resp. -14 Iπ
```

Confirmation. $-2\pi i \cdot 7$ by Cauchy's integral formula for the first term (minus because you integrate clockwise). 0 for the second term by the formula involving the first derivative (because the derivative of a constant is 0).