In [1]:

```
# To find the closest points between two parabolas
# define y=f(x) as the parabola P_1 #
f(x)=x^2
show(f)
```

Out[1]:

$$x \mapsto x^2$$

In [2]:

```
# define y=g(x) as the parabola P_2
#
g(x)=-x^2-16*x-65
show(g)
```

Out[2]:

$$x \mapsto -x^2 - 16\,x - 65$$

In [3]:

```
# fdash(x) be the derivative f'(x)
#
fdash(x)=diff(f(x),x)
show(fdash)
```

Out[3]:

$$x \mapsto 2\,x$$

In [4]:

```
# gdash(x) be the derivative g'(x)
#
gdash(x)=diff(g(x),x)
show(gdash)
```

Out[4]:

$$x \mapsto -2\,x - 16$$

In [5]:

```
# Let u be x_1, the x-coordinate of point on P_1 closest to P_2
# Let v be x_2, the x-coordinate of point on P_2 closest to P_1
# solve f'(x_1)=g'(x_2) for x_2 (in terms of x_1) # i.e. solve fdash(u)=gdash(v)
for v (in terms of u) #
var('u v')
parallel=solve(fdash(u)==gdash(v),v)
v=parallel[0].rhs()
show(v)
```

Out[5]:

$$-u-8$$

In [6]:

```
# define the square of distance between the two points # (x_1,y_1) and (x_2,y_2)
# i.e.  (u,f(u)) and (v,g(v))
#
DS(u)=(v-u)^2+(g(v)-f(u))^2
show(DS)
```

Out[6]:

$$u \mapsto \left((u+8)^2 + u^2 - 16\,u - 63\right)^2 + 4\,(u+4)^2$$

In [7]:

```
# find minimum point of DS with respect of u # i.e. solve DS'(u)==0 for u
#
show(solve(diff(DS(u),u)==0,u))
```

Out[7]:

$$\left[u = -\frac{1}{2}i\,\sqrt{7} + \frac{1}{2}, u = \frac{1}{2}i\,\sqrt{7} + \frac{1}{2}, u = (-1)\right]$$

In [8]:

```
# the only real root is the last one
#
var('u0')
u0=solve(diff(DS(u),u)==0,u)[2].rhs()
show(u0)
```

Out[8]:

$$-1$$

In [9]:

```
# check to see if second derivative is positive to ensure is a minimum
var('uu')
show(bool(diff(DS(uu),uu,2).subs(uu==u)>0))
```

Out[9]:

True

In [10]:

```
# use the value of u to find v
#
var('u1')
u1=v.subs(u==u0)
show(u1)
```

Out[10]:

$$-7$$

In [11]:

```
# find the y-coordinate y_1=f(x_1)
#
# find the y-coordinate y_2=g(x_2)
#
var('v0 v1' )
v0=f(x).subs(x==u0)
v1=g(x).subs(x==u1)
show(v0)
show(v1)
```

Out[11]:

$$1$$

Out[11]:

$$-2$$

In [12]:

```
# find the distance (square root of DS) at u
#
show(sqrt(DS(u)).subs(u=u0))
```

Out[12]:

$$3\sqrt{5}$$

In [13]:

```
# get y=ftangent(x), the tangent line of P_1 at the point (u,f(u))
#
var('y')
#ftangent(x)=solve((y-v0/(x-u0)==fdash(u).subs(x==x0),y)[0].rhs()
ftangent(x)=solve((y-v0)/(x-u0)==fdash(u0),y)[0].rhs()
show(ftangent)
```

Out[13]:

$$x \mapsto -2\,x - 1$$

In [14]:

```
# get y=gtangent(x), the tangent line of P_2 at the point (v(u),g(v(u)))
#
var('y')
#gtangent(x)=solve((y-g(v(u)))/(x-v(u))==gdash(v(u)),y)[0].rhs()
gtangent(x)=solve((y-v1)/(x-u1)==gdash(u1),y)[0].rhs()
show(gtangent)
```

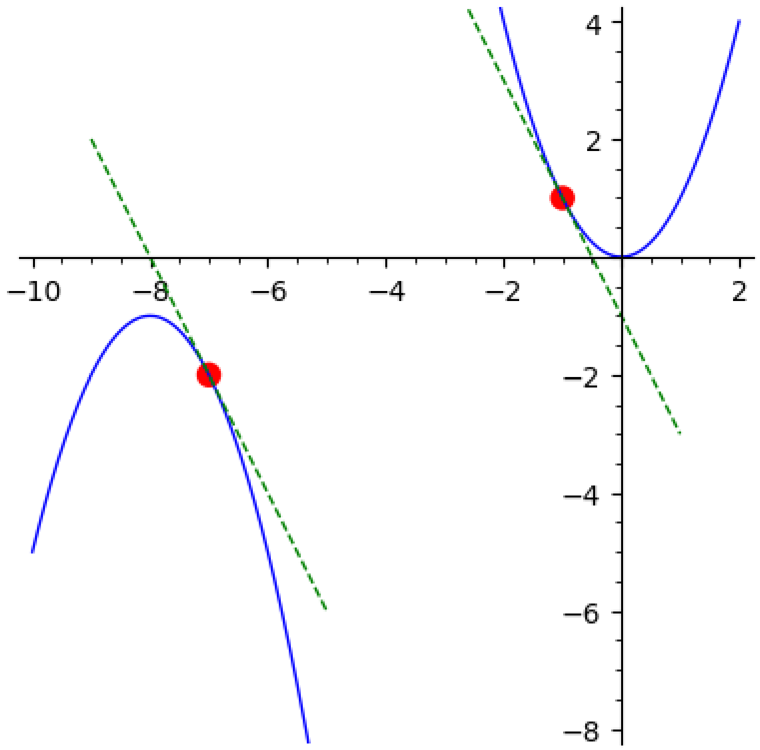
Out[14]:

$$x \mapsto -2\,x - 16$$

In [15]:

```
p1=plot(f(x),x,-10,2)
p2=plot(g(x),x,-10,2)
p3=plot(ftangent(x),x,-2.6,1, rgbcolor='green',   linestyle = "dashed")
p4=plot(gtangent(x),x,-9,-5, rgbcolor='green',   linestyle = "dashed")
pt1 = point((-1,1), rgbcolor='red', pointsize=80)
pt2 = point((-7,-2), rgbcolor='red', pointsize=80)
(p1+p2+p3+p4+pt1+pt2).show(xmin=-10, xmax=2, ymin=-8, ymax=4,aspect_ratio=1)
```

Out[15]:



In [0]: