Exam2

```
a = matrix([[1],[2],[3]])
```

```
a
```

```
[1]
[2]
[3]
```

```
a*a.transpose()
```

```
[1 2 3]
[2 4 6]
[3 6 9]
```

```
(a.transpose()*a)[0,0]
```

```
14
```

```
P = a*a.transpose() / (a.transpose()*a)[0,0]
```

```
P
```

```
[1/14  1/7 3/14]
[ 1/7  2/7  3/7]
[3/14  3/7 9/14]
```

```
P.row_space()
```

```
Vector space of degree 3 and dimension 1 over Rational Field
Basis matrix:
[1 2 3]
```

```
P.transpose().kernel()
```

```
Vector space of degree 3 and dimension 2 over Rational Field
Basis matrix:
[   1    0 -1/3]
[   0    1 -2/3]
```

```
P.kernel()
```

```
Vector space of degree 3 and dimension 2 over Rational Field
Basis matrix:
[   1    0 -1/3]
[   0    1 -2/3]
```

```
P.column_space()
```

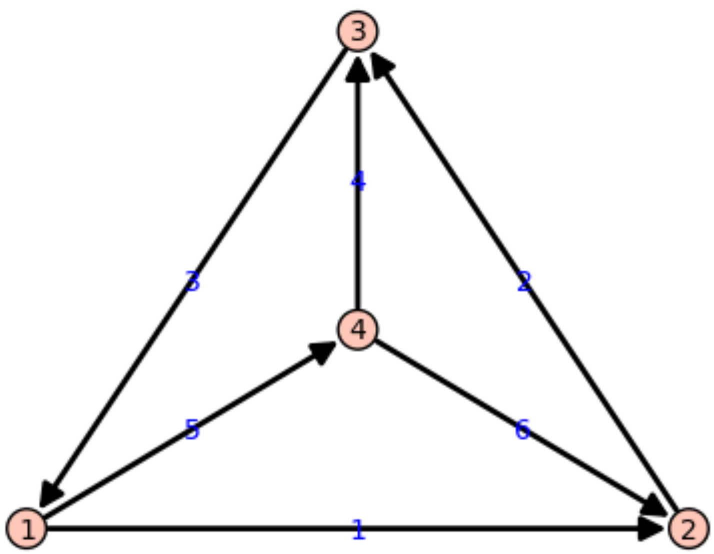
```
Vector space of degree 3 and dimension 1 over Rational Field
Basis matrix:
```

```
[1 2 3]
```

```
g = DiGraph({ 1:{2:'1',4:'5'}, 2:{3:'2'}, 3:{1:'3'},
4:{3:'4',2:'6' }})
```

```
g.set_pos( {1:[-1,0], 2:[1,0], 3:[0,1.5], 4:[0,.6]} )
```

```
g.show(edge_labels=True)
```




```
A = g.incidence_matrix().transpose()
```

```
A
```

```
[-1  0  0  1]
[-1  1  0  0]
[ 0 -1  1  0]
[ 0  0  1 -1]
[ 0  1  0 -1]
[ 1  0 -1  0]
```

```
A.column_space()
```

```
Sparse free module of degree 6 and rank 3 over Integer Ring
Echelon basis matrix:
[ 1  0  0 -1 -1  0]
```

```
[ 0  1  0  1  1 -1]
[ 0  0  1  1  0 -1]
```

```
A.rank()
```

```
3
```

```
A.transpose().kernel()
```

```
Free module of degree 4 and rank 1 over Integer Ring
Echelon basis matrix:
[1 1 1 1]
```

```
(A.transpose()*A).right_eigenvectors()
```

```
[(0, [
(1, 1, 1, 1)
], 1), (4, [
(1, 0, 0, -1),
(0, 1, 0, -1),
(0, 0, 1, -1)
], 3)]
```

```
A = matrix([ [0,1],[20,1],[40,1],[80,1] ])
A
```

```
[ 0  1]
[20  1]
[40  1]
[80  1]
```

```
b = vector([70,100,130,230])
b
```

```
(70, 100, 130, 230)
```

```
x = A.solve_right(b)
```

```
Traceback (click to the left of this block for traceback)
...
ValueError: matrix equation has no solutions
```

```
B = A.transpose() * A
B
```

```
[8400  140]
[ 140    4]
```

```
d = A.transpose() * b
d
```

```
(25600, 530)
```

```
Y = B.solve_right(d)
```

```
Y
```

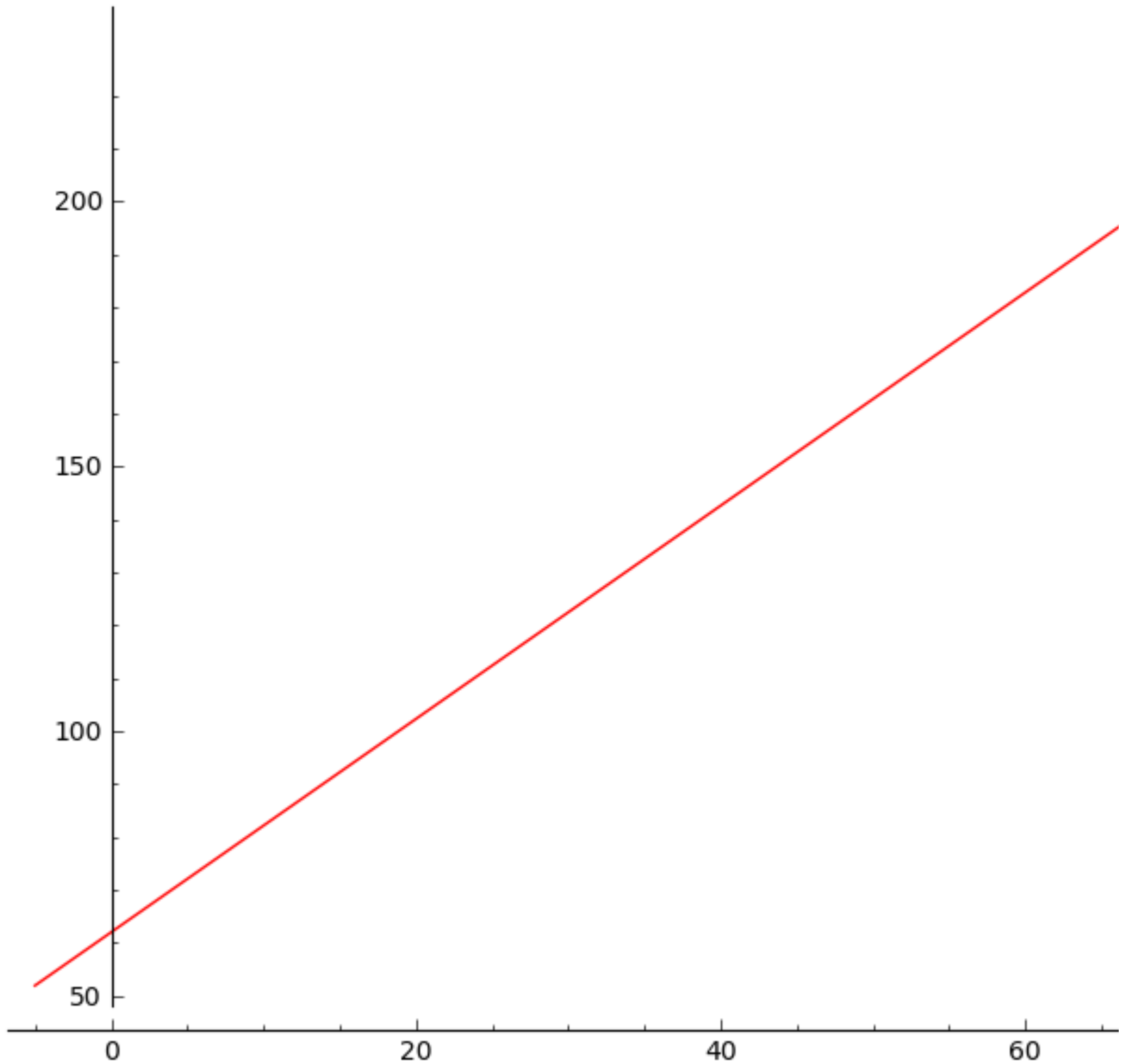
(141/70, 62)

```
y = Y[0] * x + Y[1]
y
```

141/70*x + 62

```
L = plot(y, (-5,85), color='red')
```

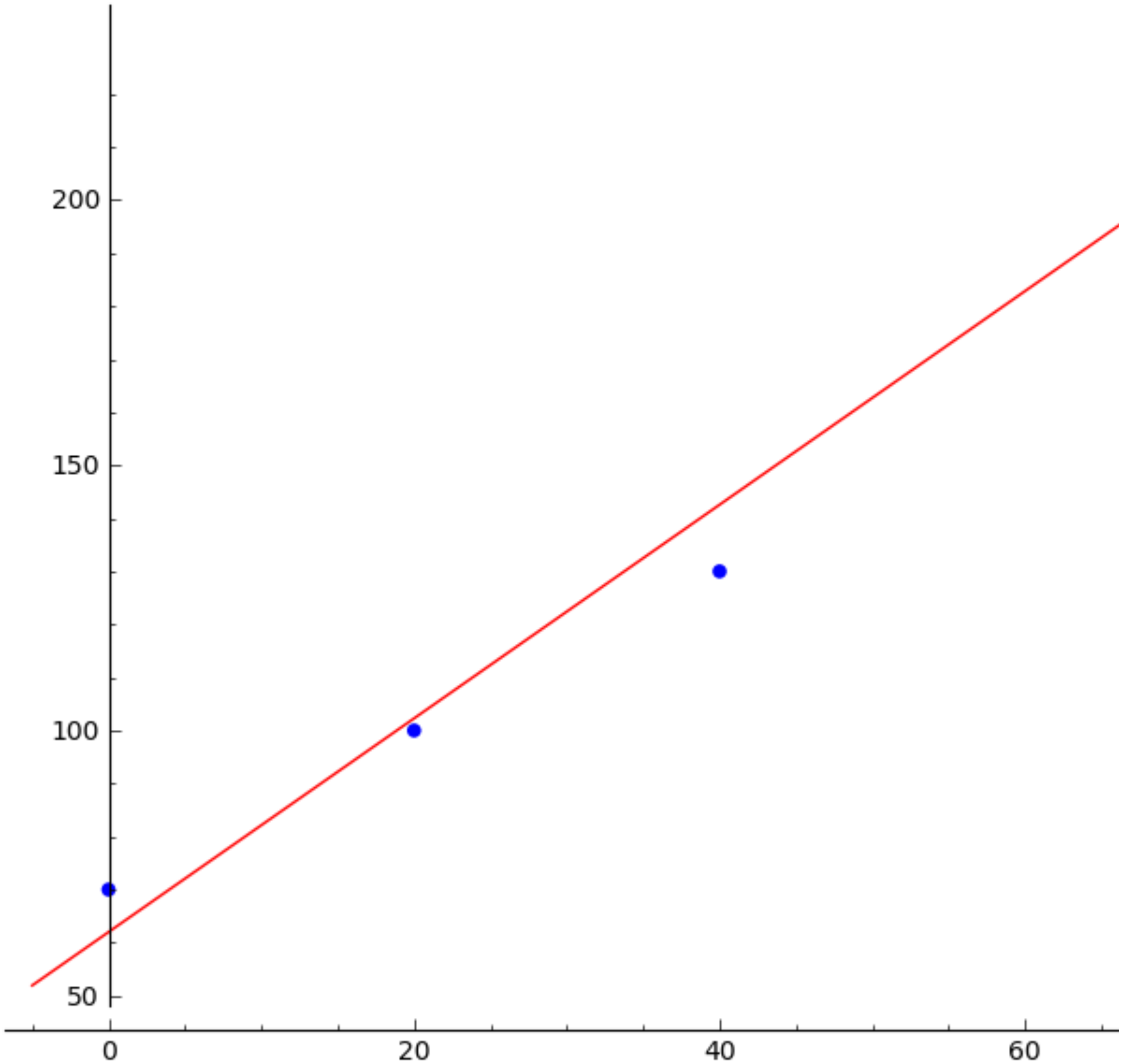
```
L
```




```
c1 = (0,70)
c2 = (20,100)
```

```
c3 = (40,130)
c4 = (80,230)
```

```
(point([c1,c2,c3,c4], size=30)+ L).show()
```




```
p1 = (0,y(x=0))
p2 = (20, y(x=20))
p3 = (40, y(x=40))
p4 = (80, y(x=80))
```

```
Vector = vector((c1[1],c2[1],c3[1],c4[1]))
```

```
Vector
```

(70, 100, 130, 230)

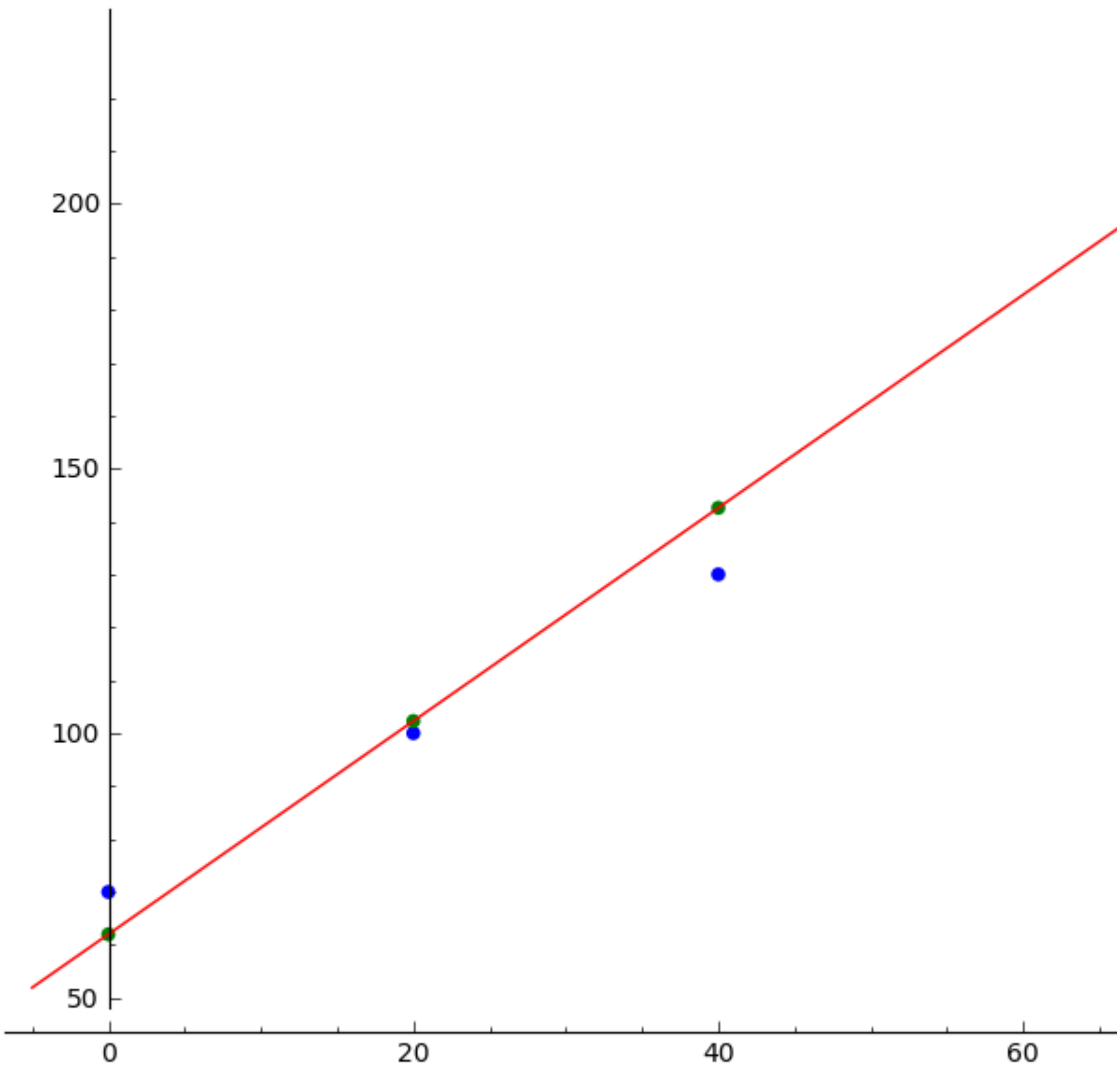
```
Projection = vector((p1[1],p2[1],p3[1],p4[1]))
Projection
```

(62, 716/7, 998/7, 1562/7)

```
Error = Vector - Projection
Error
```

(8, -16/7, -88/7, 48/7)

```
(point([c1,c2,c3,c4], size=30)+ L + point([p1,p2,p3,p4],size=30,
color = 'green')).show()
```




```
y(x=60).n(16)
```

182.9

```
y(x=1100).n(16)
```

```
2278.
```

```
M=matrix([[-1,1,0,0],[0,-1,1,0],[0,0,-1,-1]])
```

```
B=M.transpose()*M
```

```
B[0,0]=B[3,3]=2
```

```
B.determinant()
```

```
5
```

```
B
```

```
[ 2 -1  0  0]
[-1  2 -1  0]
[ 0 -1  2  1]
[ 0  0  1  2]
```