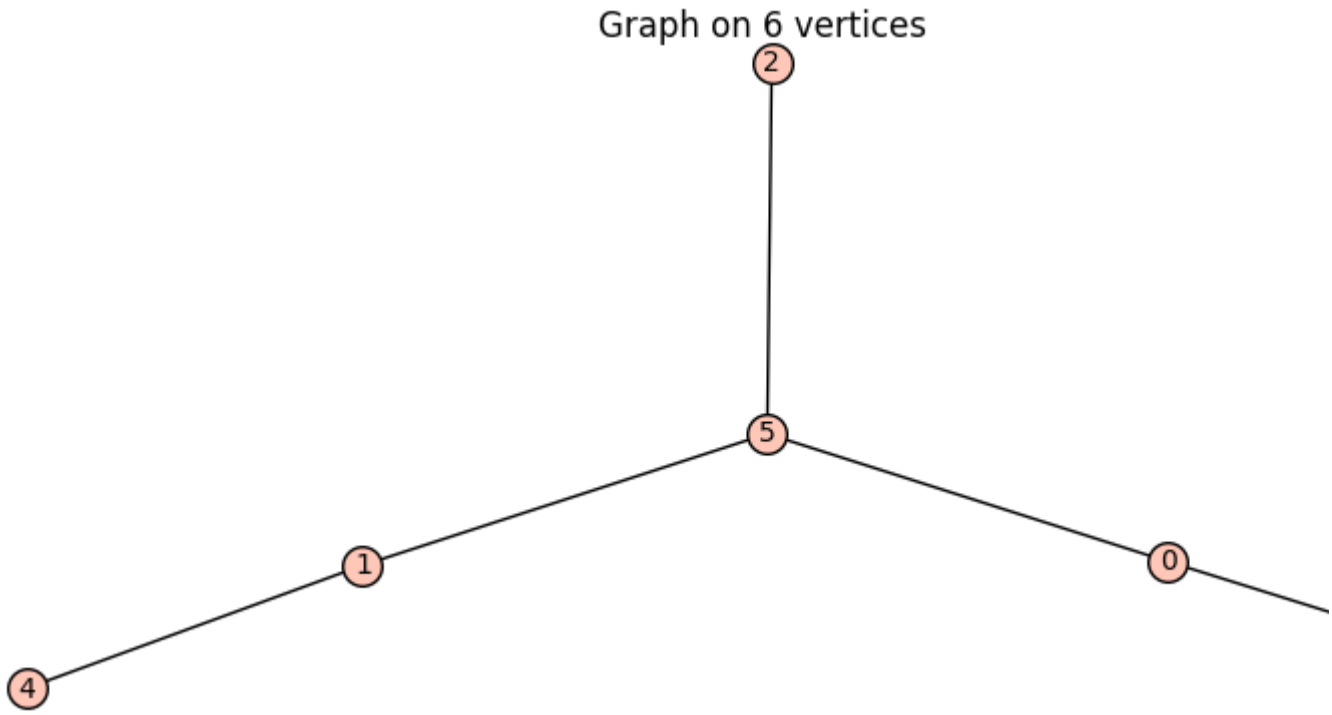
Dom

```
##### Finding graphs on 6 vertices with domination number 3
```

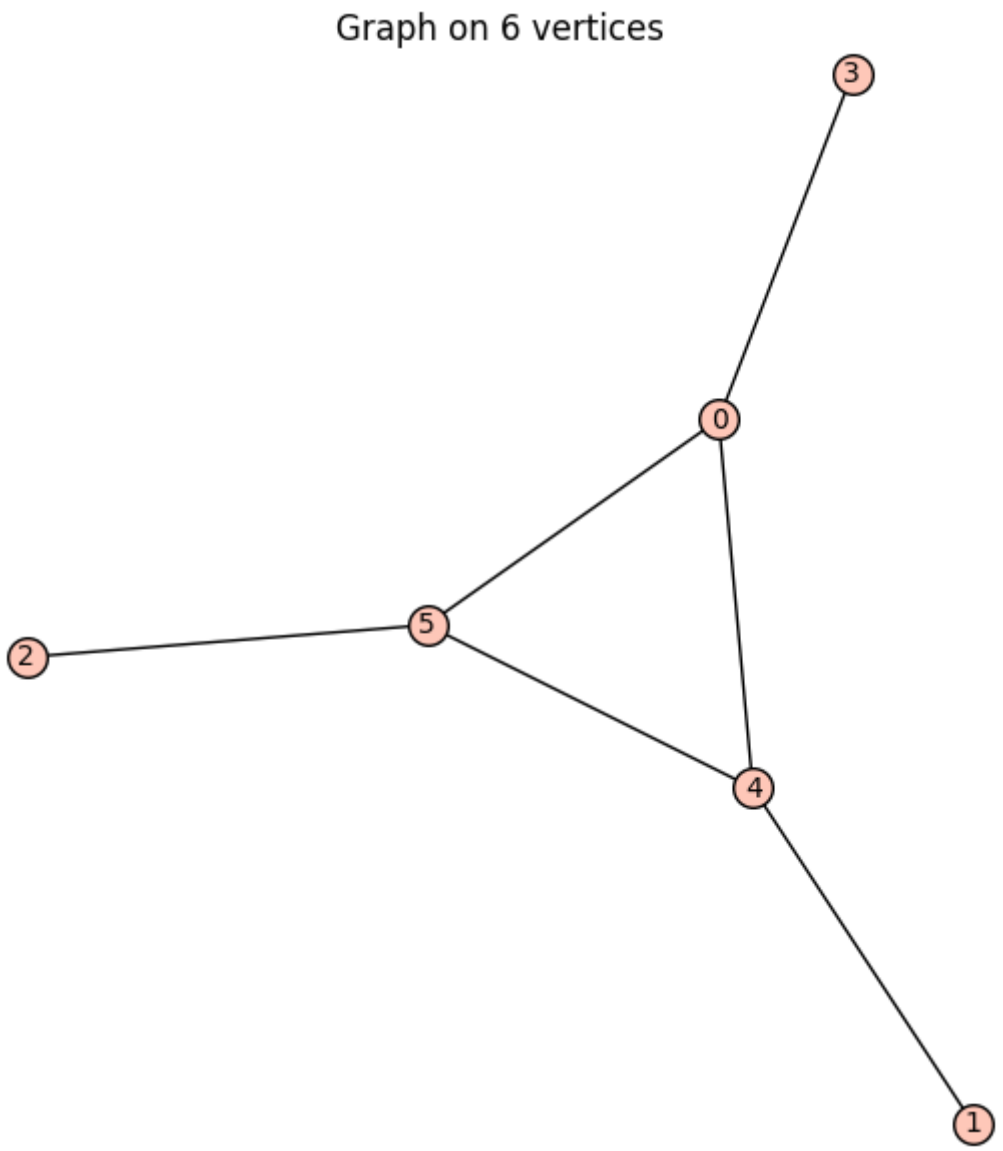
```
for x in graphs.nauty_geng(str(6)+" -c"):
    if x.dominating_set(value_only=True)==3:
        print x.graph6_string()
```

```
ECR_
ECqg
```

```
Graph('ECR_')
```




```
Graph('ECqg')
```




```
##### Finding graphs on 7 vertices with domination number 7
```

```
L=[]
for x in graphs.nauty_geng(str(7)+" -c"):
    if x.dominating_set(value_only=True)==3:
        L.append(x.graph6_string())
print len(L)
print L
```

```
42
['F?`F_', 'F?bDg', 'F?`e_', 'F?`cg', 'F?`f_', 'F?`eg', 'F?beg',
'F?q`o', 'F?q_w', 'F?qdo', 'F?qcw', 'F?ov?', 'F?ov_', 'F?re_',
'F?reg', 'F?qkw', 'FCOf?', 'FCOf_', 'FCQb?', 'FCQf?', 'FCQe_',
'FCQb_', 'FCQeO', 'FCQeG', 'FCQf_', 'FCQfG', 'FCQeW', 'FCQV_',
'FCRfG', 'FCReg', 'FCRbg', 'FCRV?', 'FCp`_', 'FCpb_', 'FCpbO',
'FCpbo', 'FCpV?', 'FCrLW', 'FCXf?', 'FCXfO', 'FCZb_', 'FCZbg']
```

```
#### Five of these graphs are trees
T=[]
for x in L:
    if Graph(x).is_tree():
```

```
        T.append(x)
print len(T)
print T
```

```
5
['F?`F_', 'F?`e_', 'F?`cg', 'FCOf?', 'FCQb?']
```

```
#### 15 of these graphs are chordal (and not trees)
C=[]
for x in L:
    if Graph(x).is_chordal():
        if Graph(x).is_tree()==False:
            C.append(x)
print len(C)
print C
```

```
15
['F?bDg', 'F?`eg', 'F?beg', 'F?qcw', 'F?qkw', 'FCOf_', 'FCQe_',
'FCQeO', 'FCQeG', 'FCQfG', 'FCQeW', 'FCQV_', 'FCRfG', 'FCReg',
'FCrLW']
```

```
### One graph is the cycle on 7 vertices
Cyc=[]
for x in L:
    if Graph(x).is_cycle():
        Cyc.append(x)
print len(Cyc)
print Cyc
```

```
1
['FCp`_']
```

```
### 21 graphs remain
Remain=[]
for x in L:
    if x not in T:
        if x not in C:
            if x not in Cyc:
                Remain.append(x)
print len(Remain)
print Remain
```

```
21
['F?`f_', 'F?q`o', 'F?q_w', 'F?qdo', 'F?ov?', 'F?ov_', 'F?re_',
'F?reg', 'FCQf?', 'FCQb_', 'FCQf_', 'FCRbg', 'FCRV?', 'FCpb_',
'FCpbO', 'FCpbo', 'FCpV?', 'FCXf?', 'FCXfO', 'FCZb_', 'FCZbg']
```

```

```