In [1]:
```
#Problem 2-2
using JuMP, Cbc, NamedArrays
items = [
    :A,
    :B,
    :C,
    :D,
    :E,
    :F
]
weight= Dict(zip(items, [5,7,10,12,2,5]))
value= Dict(zip(items, [50,20,60,65,15,70]))


m=Model()
#DVs
@variable(m, x[items]>=0)

#Objective
@objective(m, Max, sum(value[i] for i in items))

#Constraints
@constraint(m, weight[weight in items], sum(x[weight]) <=18)


set_optimizer(m, Cbc.Optimizer)
optimize!(m)
```

```
Presolve 0 (-6) rows, 0 (-6) columns and 0 (-6) elements
Optimal - objective value -0
After Postsolve, objective 0, infeasibilities - dual 0 (0), primal 0 (0)
Optimal objective 0 - 0 iterations time 0.002, Presolve 0.00
```

In [ ]: