

In [1]:

```
using JuMP, GLPK
```

In [10]:

```
c1 = [100 , 150]
A1 = [1 1 ; -1 0 ; 0 -1]
b1 = [120 ; -40 ; -20]

main = Model(GLPK.Optimizer)
@variable(main , x[1:2])
@variable(main , -1000 ≤ θ)
@objective(main , Min, c1' * x + θ)
@constraint(main, A1 * x .≤ b1)
optimize!(main)
println("#####")
println("#####")
@show value.(x) , value(θ)
@show objective_value(main)
```

```
#####
#####
(value.(x), value(θ)) = ([40.0, 20.0], -1000.0)
objective_value(main) = 6000.0
```

Out[10]:

```
3-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}, MathOptInterface.LessThan{Float64}}, ScalarShape}}:
 x[1] + x[2] ≤ 120.0
 -x[1] ≤ -40.0
 -x[2] ≤ -20.0
```

In [23]:

```
x = value.(x)
c2 = [-24 , -28]
A2 = [6 10 ; 8 5 ; 1 0 ; 0 1]
h = [-60 0 ; 0 -80 ; 0 0 ; 0 0]
b2 = [0 ; 0 ; 500 ; 100]

sub1 = Model(GLPK.Optimizer)
@variable(sub1 , y[1:2])
@objective(sub1 , Min, c2' * y)
@constraint(sub1 , A2 * y + h * x .≤ b2)
optimize!(sub1)
println("#####")
println("#####")
@show value.(y)
consts_sub1 = all_constraints(sub1 , AffExpr , MOI.LessThan{Float64})
@show dual.(consts_sub1)
@show objective_value(sub1);
```

```
#####
#####
value.(y) = [137.5, 100.0]
dual.(consts_sub1) = [0.0, -3.0, 0.0, -13.0]
objective_value(sub1) = -6100.0
```

In [29]:

```
x = value.(x)
c2 = [-28 , -32]
A2 = [6 10 ; 8 5 ; 1 0 ; 0 1]
h = [-60 0 ; 0 -80 ; 0 0 ; 0 0]
b2 = [0 ; 0 ; 300 ; 300]

sub2 = Model(GLPK.Optimizer)
@variable(sub2 , 0 ≤ y[1:2])
@objective(sub2 , Min , c2' * y)
@constraint(sub2 , A2 * y +h * x .≤ b2)
optimize!(sub2)
println("#####")
println("#####")
@show value.(y)
consts_sub2 = all_constraints(sub2 , AffExpr , MOI.LessThan{Float64})
@show dual.(consts_sub2)
@show objective_value(sub2);
```

```
#####
#####
value.(y) = [80.0, 192.0]
dual.(consts_sub2) = [-2.3200000000000003, -1.7599999999999993, 0.0, 0.0]
objective_value(sub2) = -8384.0
```

In [ ]:

