A shipping company supplies goods to three customers, who require 40, 50 and 40 units respectively. The company has three warehouses, each of which has 30 units available. The costs of shipping 1 unit from each warehouse to each customer are shown in the table below.

From/To	Customer 1	Customer 2	Customer 3
Warehouse 1	$15	$35	$25
Warehouse 2	$10	$50	$40
Warehouse 3	$20	$40	$30

There is a penalty for unmet demand: With customer 1, a penalty cost of $70 per unit is incurred; with customer 2, 75 per unit; and with customer 3, $65 per unit. The company's goal is to minimize the total cost.

In [ ]:
```
using JuMP, HiGHS

# defining model
company = Model(HiGHS.Optimizer)
```

Out[ ]:
```
A JuMP Model
Feasibility problem with:
Variables: 0
Model mode: AUTOMATIC
CachingOptimizer state: EMPTY_OPTIMIZER
Solver name: HiGHS
```

In [ ]:
```
costs = [15 35 35;
         10 50 40;
         20 40 30;
         70 75 65]
demand = [40 50 40]
supply = [30 30 30 40]

num_warehouses = 4
num_customer = 3
```

Out[ ]:
```
3
```

In [ ]:
```
#Create a variable xij for each pair of warehouse and customer that represent
#the amount of goods sent from warehouse i to customer j
@variable(company, x[1:num_warehouses,1:num_customer] >= 0)

#Each warehouse i can provide at most supply[i] goods
@constraint(company, supplyconstraint[i in 1:num_warehouses], sum(x[i,j] for

#Each customer j requires at least demand[j] goods
@constraint(company, demandconstraint[j in 1:num_customer], sum(x[i,j] for i

#Cost of sending from warehouse i to customer j is given by costs[i,j]
@objective(company, Min, sum(sum(costs[i,j]*x[i,j] for j in 1:num_customer) f
```

Out[ ]:

$$15x_{1,1} + 35x_{1,2} + 35x_{1,3} + 10x_{2,1} + 50x_{2,2} + 40x_{2,3} + 20x_{3,1} + 40x_{3,2} + 30x_{3,3} + 70x_{4,}$$

In [ ]:
```
# solving the optimization problem
optimize!(company)
```

```
Presolving model
7 rows, 12 cols, 24 nonzeros
7 rows, 12 cols, 24 nonzeros
Presolve : Reductions: rows 7(-0); columns 12(-0); elements 24(-0)
Solving the presolved LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 3(130) 0s
          8     4.9500000000e+03 Pr: 0(0) 0s
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Simplex   iterations: 8
Objective value     :  4.9500000000e+03
HiGHS run time      :          0.01
```

In [ ]:
```
# matrix representing the amount of goods sent from each warehouse to each
# customer
value.(x)
```

Out[ ]:

```
4×3 Matrix{Float64}:
  0.0  30.0   0.0
 30.0   0.0   0.0
 10.0   0.0  20.0
  0.0  20.0  20.0
```