3. ABC Investments [15 pts]

ABC Inc. is considering several investment options. Each option has a minimum investment required as well as a maximum investment allowed. These restrictions, along with the expected return are summarized in the following table (figures are in millions of dollars):

Option	Minimum investment	Maximum investment	Expected return (%)
1	3	27	13
2	2	12	9
3	9	35	17
4	5	15	10
5	12	46	22
6	4	18	12

Because of the high-risk nature of Option 5, company policy requires that **the total amount invested in Option 5 be no more that the combined amount invested in Options 2, 4 and 6.** In addition, **if an investment is made in Option 3, it is required that at least a minimum investment be made in Option 6.** ABC has $80 million to invest and obviously wants to maximize its total expected return on investment. Which options should ABC invest in, and how much should be invested?

In [1]:

```
# data
min_investments = [3, 2, 9, 5, 12, 4]
max_investments = [27, 12, 35, 15, 46, 18]
expected_returns = [13, 9, 17, 10, 22, 12]*0.01 + 1

max_investment = 80
```

In [8]:

```
using JuMP, Cbc, Gurobi, Mosek, GLPK

m = Model(solver = MosekSolver())

@variable(m, x[1:6] >= 0)
@variable(m, z[1:6], Bin)
@constraint(m, sum(x[1:6]) <= max_investment)

@constraint(m, x[1] >= min_investments[1]z[1])
@constraint(m, x[1] <= max_investments[1]z[1])

@constraint(m, x[2] >= min_investments[2]z[2])
@constraint(m, x[2] <= max_investments[2]z[2])

@constraint(m, x[3] >= min_investments[3]z[3])
@constraint(m, x[3] <= max_investments[3]z[3])

@constraint(m, x[4] >= min_investments[4]z[4])
@constraint(m, x[4] <= max_investments[4]z[4])

@constraint(m, x[5] >= min_investments[5]z[5])
@constraint(m, x[5] <= max_investments[5]z[5])
@constraint(m, x[5] <= (x[2] + x[4] + x[6]))

@constraint(m, x[6] >= min_investments[6]z[6])
@constraint(m, x[6] <= max_investments[6]z[6])
@constraint(m, z[3] <= z[6])

@expression(m, total_returns, expected_returns[1]*x[1] + expected_returns[2]*x[2] + expected_returns[3]*x[3] + expected_re

@objective(m, Max, total_returns)

solve(m)
```

Out[8]: :Optimal

In [9]:

```
println("ABC should invest as follows:")

for i = 1:6
        println("Option ", i, " with \$", getvalue(x[i]), " million invested")
end

println()
println("This brings the total investment to \$", sum(getvalue(x[1:6])), " million, with an expected return of \$", getobj
```

```
ABC should invest as follows:
Option 1 with $0.0 million invested
Option 2 with $0.0 million invested
Option 3 with $35.0 million invested
Option 4 with $5.0 million invested
Option 5 with $22.5 million invested
Option 6 with $17.5 million invested

This brings the total investment to $80.0 million, with an expected return of $93.5 million
```