In [1]:

```
using JuMP
```

In [2]:

```
m = Model()
```

Out[2]:

$$\min \quad 0$$

Subject to

In [3]:

```
foods = ["wine", "beer", "pizza", "burger", "fries", "cola", "apple", "donut"]
values = [89,90,95,100,90,79,50,10]
calories = [123,154,258,354,365,150,95,195]
```

Out[3]:

```
8-element Array{Int64,1}:
 123
 154
 258
 354
 365
 150
  95
 195
```

In [4]:

```
@variable(m, x[1:8], Bin)
```

Out[4]:

$$x_i \in \{0,1\} \quad \forall i \in \{1,2,\dots,7,8\}$$

In [5]:

```
@constraint(m, sum{x[i] * calories[i], i in 1:8} <= 750 )
```

Out[5]:

$$123 x_1 + 154 x_2 + 258 x_3 + 354 x_4 + 365 x_5 + 150 x_6 + 95 x_7 + 195 x_8 \leq 750$$

In [6]:

```
@objective(m, Max, sum{x[i]* values[i], i in 1:8})
```

Out[6]:

$$89 x_1 + 90 x_2 + 95 x_3 + 100 x_4 + 90 x_5 + 79 x_6 + 50 x_7 + 10 x_8$$

In [7]:

```
solve(m)
```

Out[7]:

```
:Optimal
```

In [8]:

```
println("Objective value: ", getobjectivevalue(m))
```

```
Objective value: 353.0
```

In [9]:

```
a = getvalue(x)
println("Foods to choose:")
for i in 1:8
    if a[i] == 1.0
        println(i,": ", foods[i])
    end
end
```

```
Foods to choose:
1: wine
2: beer
3: pizza
6: cola
```

In [ ]:

```

```