

# Super sintesi AMPL

## v 1.0

Sintesi dei comandi base per affrontare una modellazione in AMPL.

**Fonte:**

<http://www.math.unipd.it/~luigi/courses/ricop/m06.lab.03.manualeAMPL.pdf>

## 1 Struttura programma

Un programma AMPL si compone di file dei seguenti file:

- .mod - per il modello;
- .dat - per i dati separati dal modello, in modo che sia facile cambiarli;
- .run - per assemblare le parti ed eseguire lo script.

## 2 Regole generiche

Alcune regole base:

- AMPL è case sensitive;
- i commenti si scrivono con #;
- le istruzioni del modello vanno separate dal punto e virgola;
- con .. si può indicare un intervallo di valori (es: 1..5).

## 3 Elementi di un programma

### 3.1 Insiemi

Gli insiemi descrivono i domini del problema; raggruppano in un vettore i dati del problema.

Dichiarati nel model:

```
set Insieme;
```

Definiti nel .dat:

```
set Insieme := 1 2 3 4;
```

`Insieme[i]` indica il valore nella posizione i-esima.

È possibile definire insiemi non solo esplicitamente, ma anche implicitamente attraverso le **espressioni di indicizzazione**. Queste espressioni si usano per definire un insieme senza attribuirgli un nome. L'insieme viene poi usato per fare da indice in una sommatoria (o altro). In un'espressione di indicizzazione possono comparire, oltre agli insiemi componenti, anche gli indici che li scorrono.

```
sum{j in Cibi: Costi[j] >= 10} X[j];
```

L'istruzione somma i valori del vettore `X` corrispondenti agli indici nell'insieme `Cibi` i cui corrispondenti valori nel vettore `Costi` sono non inferiori a 10.

### 3.2 Dati

I dati sono i valori numerici che definiscono in dettaglio il problema che si vuole affrontare, una volta precisata la sua struttura logica. Dichiarati nel model:

```
param N;  
set Cibi;  
set Nutrienti;  
param Costi{Cibi};
```

Definiti nel .dat:

```

param N := 10;

#assegna ad ogni valore dell'insieme un valore dato
#(in questo caso un numero al cibo)
set Cibi := Pane Latte Carne Verdure;
param Costi := Pane 10
Latte 5
Carne 20
Verdure 8;

```

### 3.3 Variabili

Le variabili vengono dichiarate nel file del modello attraverso la parola chiave `var`. Possono avere delle restrizioni già della definizione.

```

var x >=0, <=1000;

# variabili con espressioni indicizzate
set Cibi;
var Consumo{Cibi};
var Quantita{j in Cibi} >= 0, <= QuantitaMax[j];

```

### 3.4 Funzione obiettivo

La funzione viene indicata nel model da:

```

minimize/maximize Nome : calcolaQualcosa;

# esempio
minimize CostoTotale : sum{j in Cibi} Costi[j] * Quantita[j];

```

### 3.5 Vincoli

I vincoli si definiscono nel model con:

```

subject to/s.t. Nome : espressione;

# esempio
subject to Fabbisogno{i in Nutrienti}:
sum{j in Cibi} A[i,j] * Quantita[j] >= Richiesta[i];

```

## 4 Esempio esercizio - Caseificio

```
#Caseificio.mod - File del modello AMPL - Caseificio

#### DICHIARAZIONE INSIEMI, PARAMETRI E VARIABILI ####
set PROD;
# insieme dei prodotti
param d{PROD}>=0;
# domanda massima per i prodotti
param l{PROD}>=0;
# quantita' di latte richiesta per ogni Kg di prodotto
param v{PROD}>=0;
# prezzo di vendita dei prodotti
param q{PROD}>=0;
# massima qt di ciascun prodotto fabbricata da 1 operaio
param N;
# numero di operai disponibili
param L;
# quantita di latte disponibile
var x{PROD} >=0;
# Kg prodotti per ogni tipo di prodotto

#### FUNZIONE E VINCOLI ####
maximize guadagno_totale: sum{i in PROD} v[i]*x[i];
subject to domanda{i in PROD}: x[i]<=d[i];
# vincolo di domanda
subject to disp_latte: sum{i in PROD} l[i]*x[i]<= L;
# vincolo sulla disponibilita del latte
subject to disp_operai: sum{i in PROD} x[i]/q[i]<=N;
# vincolo sulla disponibilita degli operai

#Caseificio.dat - File dei dati - Caseificio

#### DEFINIZIONE INSIEMI, PARAMETRI E VARIABILI ####
set PROD:= Burro, Ricotta, Mozzarella;
param: d l v q :=
Burro 9 20 2.5 2
```

```
Ricotta 23 2 1.1 7  
Mozzarella 18 5 2 3 ;  
param N:=7;  
param L:=120;
```