

# LINGO - [www.lindo.com](http://www.lindo.com)

- Linear

# LINGO - [www.lindo.com](http://www.lindo.com)

- Linear
- INteractive and

# LINGO - [www.lindo.com](http://www.lindo.com)

- Linear
- INteractive and
- General

# LINGO - [www.lindo.com](http://www.lindo.com)

- Linear
- INteractive and
- General
- Optimizer

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “.”

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “:”

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “:”
- un commento può anche esser intercalato fra le istruzioni;



# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “;”
- un commento può anche esser intercalato fra le istruzioni;
- colori del testo:

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “;”
- un commento può anche esser intercalato fra le istruzioni;
- colori del testo:
  - blu: parole-chiave

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “:”
- un commento può anche esser intercalato fra le istruzioni;
- colori del testo:
  - blu: parole-chiave
  - verde: commenti

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “:”
- un commento può anche esser intercalato fra le istruzioni;
- colori del testo:
  - blu: parole-chiave
  - verde: commenti
  - nero: altro testo

# Programmare in LINGO

- Un programma in LINGO consiste in una serie di istruzioni;
- ogni istruzione deve terminare con “;”
- è possibile inserire commenti: essi iniziano con “!” e terminano con “:”
- un commento può anche esser intercalato fra le istruzioni;
- colori del testo:
  - blu: parole-chiave
  - verde: commenti
  - nero: altro testo
- non c'è differenza fra carattere minuscolo e maiuscolo

# Programmare in LINGO

- Ad un programma in LINGO si può assegnare un titolo tramite il comando “title” (opzionale);

# Programmare in LINGO

- Ad un programma in LINGO si può assegnare un titolo tramite il comando “title” (opzionale);
- un programma in LINGO può essere concluso con il comando “end” (opzionale); tutto ciò che viene scritto dopo il comando “end” viene ignorato;

# Programmare in LINGO

- Ad un programma in LINGO si può assegnare un titolo tramite il comando “title” (opzionale);
- un programma in LINGO può essere concluso con il comando “end” (opzionale); tutto ciò che viene scritto dopo il comando “end” viene ignorato;
- a funzione obiettivo e vincoli si può assegnare un nome (da scrivere fra parentesi quadre).



# Le variabili decisionali in LINGO

- I nomi delle variabili decisionali devono cominciare sempre con un carattere alfabetico (seguito da caratteri alfabetici o numerici, o underscore, fino a un massimo di 32)

# Le variabili decisionali in LINGO

- I nomi delle variabili decisionali devono cominciare sempre con un carattere alfabetico (seguito da caratteri alfabetici o numerici, o underscore, fino a un massimo di 32)
- per default, le variabili decisionali in LINGO sono  $\geq 0$

# Le variabili decisionali in LINGO

- I nomi delle variabili decisionali devono cominciare sempre con un carattere alfabetico (seguito da caratteri alfabetici o numerici, o underscore, fino a un massimo di 32)
- per default, le variabili decisionali in LINGO sono  $\geq 0$
- variabile  $x$  libera in segno: @free(x);

# Le variabili decisionali in LINGO

- I nomi delle variabili decisionali devono cominciare sempre con un carattere alfabetico (seguito da caratteri alfabetici o numerici, o underscore, fino a un massimo di 32)
- per default, le variabili decisionali in LINGO sono  $\geq 0$
- variabile  $x$  libera in segno: @free(x);
- variabile  $x$  intera: @gin(x);

# Le variabili decisionali in LINGO

- I nomi delle variabili decisionali devono cominciare sempre con un carattere alfabetico (seguito da caratteri alfabetici o numerici, o underscore, fino a un massimo di 32)
- per default, le variabili decisionali in LINGO sono  $\geq 0$
- variabile  $x$  libera in segno: @free(x);
- variabile  $x$  intera: @gin(x);
- variabile  $x$  binaria: @bin(x);

# La funzione obiettivo in LINGO

- Massimizzare:  $\max =$

# La funzione obiettivo in LINGO

- Massimizzare:  $\max =$
- minimizzare:  $\min =$

# I modelli in LINGO

- Modello semplice



# I modelli in LINGO

- Modello semplice
- modello complesso

# I modelli complessi

- Sono modelli scritti tramite la definizione di insiemi (SETS) e dati (DATA)

# I modelli complessi

- Sono modelli scritti tramite la definizione di insiemi (SETS) e dati (DATA)
- gli insiemi possono essere:

# I modelli complessi

- Sono modelli scritti tramite la definizione di insiemi (SETS) e dati (DATA)
- gli insiemi possono essere:
  - primitivi

# I modelli complessi

- Sono modelli scritti tramite la definizione di insiemi (SETS) e dati (DATA)
- gli insiemi possono essere:
  - primitivi
  - derivati: definiti cioè a partire da altri insiemi (primitivi o derivati)

# SETS

- ```
SETS:  
    SET_NAME / MEMBER_LIST /:  
    ATTRIBUTE_LIST;  
ENDSETS
```

# SETS

- ```
SETS:  
    SET_NAME / MEMBER_LIST /:  
    ATTRIBUTE_LIST;  
ENDSETS
```
- MEMBER\_LIST e ATTRIBUTE\_LIST sono campi opzionali

# SETS

- ```
SETS:  
    SET_NAME / MEMBER_LIST /:  
    ATTRIBUTE_LIST;  
ENDSETS
```
- MEMBER\_LIST e ATTRIBUTE\_LIST sono campi opzionali
- MEMBER\_LIST:



# SETS

- ```
SETS:  
    SET_NAME / MEMBER_LIST /:  
    ATTRIBUTE_LIST;  
ENDSETS
```
- MEMBER\_LIST e ATTRIBUTE\_LIST sono campi opzionali
- MEMBER\_LIST:
  - implicita (esempio: 1..7)

# SETS

- ```
SETS:  
    SET_NAME / MEMBER_LIST /:  
    ATTRIBUTE_LIST;  
ENDSETS
```
- MEMBER\_LIST e ATTRIBUTE\_LIST sono campi opzionali
- MEMBER\_LIST:
  - implicita (esempio: 1..7)
  - esplicita (inserendo il nome di ciascun membro, uno per uno)

# DATA

- ```
DATA:  
    ATTRIBUTE_LIST = VALUE_LIST;  
ENDDATA
```

# Function SUM

- @SUM ( SET\_NAME ( SET\_INDEX\_LIST )  
| CONDITION : EXPRESSION )

# Function FOR

- @FOR ( SET\_NAME ( SET\_INDEX\_LIST )  
| CONDITION : EXPRESSION )

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:
  - #EQ# =



# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:
  - #EQ# =
  - #NE# ≠

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:
  - #EQ# =
  - #NE# ≠
  - #GE# ≥

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:
  - #EQ# =
  - #NE# ≠
  - #GE# ≥
  - #GT# >

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:
  - #EQ# =
  - #NE# ≠
  - #GE# ≥
  - #GT# >
  - #LT# <

# Il campo | CONDITION

- Il campo | CONDITION è un campo opzionale
- In CONDITION possono apparire i seguenti operatori logici:
  - #EQ# =
  - #NE# ≠
  - #GE# ≥
  - #GT# >
  - #LT# <
  - #LE# ≤

# Interazione con programmi o file esterni

Un programma LINGO può

- leggere i dati da un file di testo;

# Interazione con programmi o file esterni

Un programma LINGO può

- leggere i dati da un file di testo;
- scrivere i risultati su un file di testo;

# Interazione con programmi o file esterni

Un programma LINGO può

- leggere i dati da un file di testo;
- scrivere i risultati su un file di testo;
- leggere i dati da un file Excel;



# Interazione con programmi o file esterni

Un programma LINGO può

- leggere i dati da un file di testo;
- scrivere i risultati su un file di testo;
- leggere i dati da un file Excel;
- essere richiamato da un file Excel come risolutore (tramite la definizione di una macro).