

LIMONATA

```
!Profitto;
MAX = 2* Limonata + 1.5 * Succo;

!Vincoli di nonnegatività;
Limonata >= 0;
Succo >= 0;

!Vincolo di disponibilità;
Limonata + Succo <=4;
!Vincolo sui limoni;
30 * Limonata + 10 * Succo <= 90;
!Vincolo sullo zucchero;
Limonata + 2 * Succo <= 6;
```

Pubblicità LINGO

```
!Max numero clienti;
MAX = 400000 * x1 + 900000 * x2 + 500000 * x3 + 200000 * x4;

! Vincolo budget;
40 * x1 + 75 * x2 + 30 * x3 + 15 * x4 <= 800;

! Vincolo clienti donne;
300 * x1 + 400 * x2 + 200 * x3 + 100 * x4 >= 2000;

! Vincolo pubblicità in TV;
40 * x1 + 75 * x2 <= 500;

! Vincolo c1);
x1 >= 3;
! Vincolo c2);
x2 >= 2;

! Vincolo d1);
5 <= x3;
x3<= 10;

!Vincolo d2);
5 <= x4;
x4 <= 10;
```

Coloranti LINGO

```
!Profitto;
MAX = 7* C1 + 10 * C2;

!Vincoli di nonnegatività;
C1 >= 0;
C2 >= 0;

!Vincolo di disponibilità di P1;
C1 + C2 <= 750;
!Vincolo di disponibilità di P2;
C1 + 2 * C2 <= 1000;
!Vincolo di disponibilità di P1;
```

C2 <= 400;

Whisky LINGO

!Massimo Profitto;

MAX = 2 * x2 + 3 * x3 - x4 + x5 + 2 * x6;

! Vincoli di non negatività;

x1 >= 0;

x2 >= 0;

x3 >= 0;

x4 >= 0;

x5 >= 0;

x6 >= 0;

! Vincoli di disponibilità;

x1 + x4 <= 2000;

x2 + x5 <= 2500;

x3 + x6 <= 1200;

! Vincoli di composizione;

- 0.4 * x1 + 0.6 * x2 + 0.6 * x3 <= 0;

- 0.2 * x1 - 0.2 * x2 + 0.8 * x3 <= 0;

- 0.6 * x4 - 0.6 * x5 + 0.4 * x6 <= 0;

- 0.85 * x4 + 0.15 * x5 + 0.15 * x6 <= 0;

Batterie LINGO

!Profitto;

MAX = 13 * x1 + 9 * x2 + 16 * x3;

!Vincoli di nonnegatività;

x1 >= 0;

x2 >= 0;

x3 >= 0;

!Vincoli;

x2 + 2 * x3 <= 4000;

x1 >= 2 * x2;

x1 <= x3;

Candele LINGO

! Profitto massimo;

MAX = 5 * bianche + 6 * blu;

!Vincoli di nonnegatività;

bianche >= 0;

blu >= 0;

!Vincolo di tempo;

18 *bianche + 20 * blu <= 480;

!Vincoli di esperienza;

@BND(10,bianche,15);

@BND(8,blu,12);

PLI LINGO

Senza vincoli di interezza

!Problema di massimizzazione;

MAX = 7* x1 + 6.2 * x2;

!Vincoli di nonnegatività;

x1 >= 0;

x2 >= 0;

!Vincolo 1;

5 * x1 + 4 * x2 <= 80;

!Vincolo 2;

4 * x1 + 5 * x2 <= 80;

Con vincoli di interezza

!Problema di massimizzazione;

MAX = 7* x1 + 6.2 * x2;

!Vincoli di nonnegatività;

x1 >= 0;

x2 >= 0;

!Vincoli di interezza;

@GIN(x1);

@GIN(x2);

!Vincolo 1;

5 * x1 + 4 * x2 <= 80;

!Vincolo 2;

4 * x1 + 5 * x2 <= 80;

PLI0-1 Turista

!Profitto massimo;

MAX = 150* x1 + 160 * x2 + 20 * x3 + 60 * x4 + 65 * x5 + 200 * x6 + 40 * x7 + 15 * x8 + 40 * x9;

!Variabili binarie;

@BIN(x1);

```

@BIN(x2);
@BIN(x3);
@BIN(x4);
@BIN(x5);
@BIN(x6);
@BIN(x7);
@BIN(x8);
@BIN(x9);

!Vincolo di capacità;
9 * x1 + 50 * x2 + 7 * x3 + 27 * x4 + 30 * x5 + 153 * x6 + 39 * x7 + 24 * x8 + 73
* x9 <= 300;

In forma compatta
!max{150* x1 + 160 * x2 + 20 * x3 + 60 * x4 + 65 * x5 + 200 * x6 + 40 * x7 + 15 *
x8 + 40 * x9};
!x1,..., x9 in 0-1;
!9 * x1 + 50 * x2 + 7 * x3 + 27 * x4 + 30 * x5 + 153 * x6 + 39 * x7 + 24 * x8 + 73
* x9 <= 300;

SETS:
!tipo di variabili: nove;

TC/ T1, T2, T3, T4, T5, T6, T7, T8, T9/;
peso(TC):w;
profitto(TC):p;
!
assign(JOB,JOB): x_ik;
numero(TC):x;
ENDSETS

!funzione obiettivo;
MAX=@SUM(TC(i):p(i)*x(i));

!Vincolo di binarietà;
@FOR(TC(i):
@BIN(x(i)));

!Vincolo di capacità;
@SUM(TC(i):
w(i)*x(i))<=hdisp;

DATA:
p=150, 160, 20, 60, 65, 200, 40, 15, 40;
w=9, 50, 7, 27, 30, 153, 39, 24, 73;
hdisp=300;
ENDDATA

end

```

Call Center - LINGO

```
!Minimizzare il numero di persone;
```

MIN=x1+ x2 + x3 + +x4 + x5+ x6 +x7;

!Vincoli di nonnegatività;

x1>=0;

x2>=0;

x3>=0;

x4>=0;

x5>=0;

x6>=0;

x7>=0;

!Vincoli di interezza;

@GIN(x1);

@GIN(x2);

@GIN(x3);

@GIN(x4);

@GIN(x5);

@GIN(x6);

@GIN(x7);

!Vincoli;

x1 + x4 +x5+ x6+ x7 >=17;

x1 + x2+ x5 +x6 + x7 >=13;

x1 + +x2 + +x3 +x6 +x7 >=15;

x1 + x2 + +x3 + x4 + x7 >= 19;

x1 + x2 + x3 + x4 + x5 >= 14;

x2 + x3 + x4 + x5 + x6 >= 16;

x3 + x4 + x5 + x6 + x7 >=11;