

## Soluções Laboratório 1

### Questão 0.1

O sistema em Mathprog é

```
/* Variáveis de Decisão */
var p integer >=0; #placas
var c integer >=0; #canos

/* Função Objetivo */
maximize lucro: 25*p + 30*c;

/* Restrições */
s.t. taxa_prod: 7*p + 10*c <= 56000;
s.t. demanda_p: p <= 6000;
s.t. demanda_c: c <= 4000;

end;
```

e o GLPK responde com

```
/* ***** /
Objective:  obj = 192000 (MAXimum)
/* ***** /
```

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	r.3	NU	56000		56000	3

  

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	p	NU	6000	0	6000	4
2	c	B	1400	0	4000	

```
/* ***** /
```

### Questão 0.2

Sejam  $a_1, a_2, \dots, c_2, c_3$  os números de bilhetes do tipo A,B e C nos trechos 1,2 e 3, respectivamente. O sistema em mathprog juntando dados com o modelo é:

```
/* ***** /
#Voos
#1: Pelotas – Porto Alegre
#2: Porto Alegre – Torres
#3: Pelotas – Torres

/* Variáveis de Decisão */
var a1 integer >=0, <=4; #Número de bilhetes vendidos do tipo A no voo 1
var a2 integer >=0, <=8; #Número de bilhetes vendidos do tipo A no voo 2
var a3 integer >=0, <=3; #Número de bilhetes vendidos do tipo A no voo 3
var b1 integer >=0, <=8; #Número de bilhetes vendidos do tipo B no voo 1
var b2 integer >=0, <=13; #Número de bilhetes vendidos do tipo B no voo 2
var b3 integer >=0, <=10; #Número de bilhetes vendidos do tipo B no voo 3
var c1 integer >=0, <=22; #Número de bilhetes vendidos do tipo C no voo 1
var c2 integer >=0, <=20; #Número de bilhetes vendidos do tipo C no voo 2
var c3 integer >=0, <=18; #Número de bilhetes vendidos do tipo C no voo 3

/* Função Objetivo */
```

```

maximize lucro: 600*a1 + 320*a2 + 720*a3 +
                440*b1 + 260*b2 + 560*b3 +
                200*c1 + 160*c2 + 280*c3;

/* Restrições */
s.t. limite_escala_13:
    a1 + b1 + c1 + a3 + b3 + c3 <= 30;

s.t. limite_escala_23:
    a2 + b2 + c2 + a3 + b3 + c3 <= 30;

end;
/*****
E com dados separados do model é:
*****/
#Voos
#1: Pelotas – Porto Alegre
#2: Porto Alegre – Torres
#3: Pelotas – Torres

/* Conjuntos */

set V; #Conjunto de voos
set T; #Conjunto de tipos de bilhetes

/* Parâmetros */

param      Preco {i in T, j in V}; #Preço do bilhete de tipo i no voo j
param Previsao {i in T, j in V}; #Previsão de bilhetes do tipo i no voo j

/* Variáveis de Decisão */
var x {i in T, j in V} integer >=0, <= Previsao[i,j]; #Número de bilhetes vendidos do t

/* Função Objetivo */
maximize lucro:
    sum{i in T, j in V} Preco[i,j]*x[i,j];

/* Restrições */
s.t. limite_escala_13:
    sum{i in T} (x[i,1]+x[i,3]) <= 30;

s.t. limite_escala_23:
    sum{i in T} (x[i,2]+x[i,3]) <= 30;

/* Área de dados ----- */
data;

set V := 1 2 3;
set T := 'A' 'B' 'C';

param Preco :      1      2      3 :=

```

```

'A'      600  320  720
'B'      440  260  560
'C'      200  160  280;

param Previsao : 1      2      3 :=
'A'          4      8      3
'B'          8     13     10
'C'         22     20     18;

end;
/*****/

```

### Questão 0.3

O modelo de fluxo é:

```

/*****/
set VERTICES;
set ARCS within (VERTICES cross VERTICES);

param capacity{ARCS};
param weight{ARCS};
param demand{VERTICES} default 0;

var x{(i,j) in ARCS} >= 0;

minimize cost: sum{(i,j) in ARCS} x[i,j]*weight[i,j];

s.t. CAP {(i,j) in ARCS}: x[i,j]<=capacity[i,j];

s.t. BALANCE{i in VERTICES}:
    sum{j in VERTICES: (i,j) in ARCS} x[i,j]
    - sum{j in VERTICES: (j,i) in ARCS} x[j,i]
    = demand[i];

end;
/*****/

```

Crie um arquivo com os dados de entrada (coloque o mesmo em um arquivo separado!) e resolva o problema conforme especificado no trabalho.