

المادة: 402 قصر
تطبيقات الحاسب الآلي في الاقتصاد الزراعي
الأسبوع العاشر

أ.د سفر حسين القحطاني
د. بدر الدين سفيان

مثال حل مشكلة نقل باستخدام البرنامج LINGO

الكميات المطلوبة والمعروضة وتكلفة نقل الوحدة الواحدة

الطلب / العرض	C1	C2	C3	C4	العرض
WH1	6	2	6	7	30
WH2	4	9	5	3	25
WH3	8	8	1	5	21
الطلب	15	17	22	12	

لاحظ أن في هذا المثال تزيد مجموع الكميات المعروضة عن إجمالي كمية الطلب.

مكونات وأقسام ملف المشكلة للبرنامج

1. بداية ونهاية المشكلة بالأمر **Model** للبداية و الأمر **End** للنهاية.
2. قسم تعريف المتغيرات و أسماء المعاملات في مجموعات **.SETS**.
3. قسم صياغة المشكلة بلغة **Lingo**، ويحتوي هذا القسم على قيود الطلب والعرض ودالة الهدف.
4. القسم الأخير وهو قسم البيانات للمعاملات والجانب الأيمن للمعادلات (الكميات المطلوبة والمعروضة في هذا المثال).

قسم تعريف المتغيرات و أسماء المعاملات في مجموعات SETS

!A 3 Warehouse, 4 Customer
Transportation Problem;

SETS:

WAREHOUSE / WH1, WH2, WH3/ : CAPACITY;

CUSTOMER / C1, C2, C3, C4/ : DEMAND;

ROUTE(WAREHOUSE, CUSTOMER) : COST, VOLUME;

ENDSETS

1. يمكن كتابة التعليق باستخدام علامة التعجب في أول السطر. يلاحظ أن بداية أمر المجموعات sets تبعه (:), نهاية هذا الأمر هو نهاية المجموعات .endsets.
2. أسم بيان الكمية المعروضة هو الطاقة الممكنة **CAPACITY** وأسم الكمية المطلوبة **.DEMAND**.
3. أما بيان التكلفة والكمية المطلوب نقلها لكل خط سير **ROUTE** هو **COST** و **VOLUME**.

قسم صياغة المشكلة بلغة Lingo

! The objective;

```
[OBJ] min = @SUM( ROUTE: COST * VOLUME);
```

يلاحظ أن صياغة دالة الهدف مطابقة لأسلوب قيمة مجموع حاصل ضرب تكلفة نقل الوحدة Cost في الكمية المنقولة Volume لكل خط سير. وأيضاً يمكن إعطاء أسم لسطر دالة الهدف بين قوسين مربعين [obj].

قسم صياغة المشكلة بلغة Lingo

! The demand constraints;

```
@FOR( CUSTOMER( J): [DEM]
```

```
@SUM( WAREHOUSE( I): VOLUME( I, J)) >=  
DEMAND( J));
```

وبالرجوع إلى طريقة تعريف مجموعة التكلفة و الكمية المنقولة:

```
ROUTE( WAREHOUSE, CUSTOMER) : COST, VOLUME;
```

يلاحظ أنه تم تحديد اتجاهين لكل بيان للعرض Warehouse و للطلب Coustomer.

ويستخدم الأمر @for لتكرار التالي له بين قوسين لكل مكان طلب، وأعطي قيود هذه الأسطر أسم الطلب [dem]. ومن ثم فإن القيد يعني أن إجمالي الكمية المنقولة إلى مكان الطلب تساوي الكمية المطلوبة أو تزيد عنها.

قسم صياغة المشكلة بلغة Lingo

! The supply constraints;

```
@FOR( WAREHOUSE( I): [SUP]
```

```
@SUM( CUSTOMER( J): VOLUME( I, J)) <=  
CAPACITY( I));
```

وبالمثل عند صياغة قيود العرض استخدم الأمر @for لتكرار التالي له بين قوسين لكل مصدر عرض، وأعطي قيود هذه الأسطر أسم العرض [sup]. ومن ثم فإن القيد يعني أن إجمالي الكمية المنقولة من كل مصدر عرض أقل من أو تساوي الكمية المعروضة والمتاحة من هذا المصدر.

قسم البيانات للمعاملات والجانب الأيمن للمعادلات

!Here are the parameters;

DATA:

```
CAPACITY = 30, 25, 21 ;  
DEMAND = 15, 17, 22, 12;  
COST = 6, 2, 6, 7,  
        4, 9, 5, 3,  
        8, 8, 1, 5;
```

ENDDATA

يلاحظ أن بيان الجانب الأيمن للكميات المطلوب أو الكميات المعروضة على شكل مصفوفة من عمود واحد، أما معاملات الكميات المنقولة فهي على شكل مصفوفة أيضاً ولكل في اتجاهين (من المصدر ، إلى الطلب). وبالتالي تكتمل الصياغة للمشكلة في التالي:

MODEL:

**! A 3 Warehouse, 4 Customer
Transportation Problem;**

SETS:

WAREHOUSE / WH1, WH2, WH3/ : CAPACITY;

CUSTOMER / C1, C2, C3, C4/ : DEMAND;

ROUTES(WAREHOUSE, CUSTOMER) : COST, VOLUME;

ENDSETS

! The objective;

[OBJ] min = @SUM(ROUTES: COST * VOLUME);

! The demand constraints;

@FOR(CUSTOMER(J): [DEM]

@SUM(WAREHOUSE(I): VOLUME(I, J)) >=
DEMAND(J));

! The supply constraints;

@FOR(WAREHOUSE(I): [SUP]

@SUM(CUSTOMER(J): VOLUME(I, J)) <=
CAPACITY(I));

! Here are the parameters;

DATA:

CAPACITY = 30, 25, 21 ;

DEMAND = 15, 17, 22, 12;

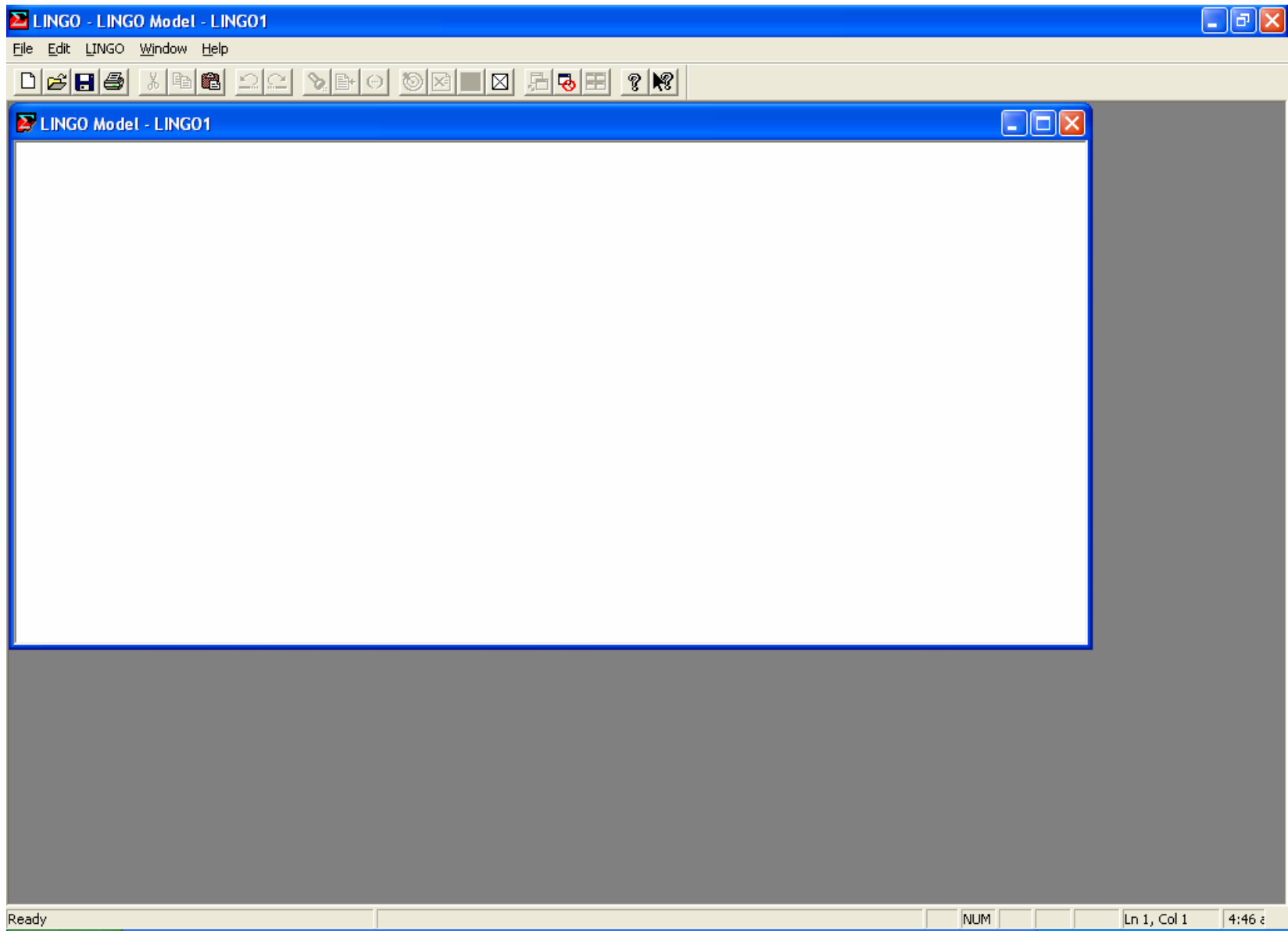
COST = 6, 2, 6, 7,

4, 9, 5, 3,

8, 8, 1, 5;

ENDDATA

END



```
LINGO - LINGO Model - LINGO1
File Edit LINGO Window Help
[Icons]
LINGO Model - LINGO1
MODEL:
! A 3 Warehouse, 4 Customer
  Transportation Problem;
SETS:
  WAREHOUSE / WH1, WH2, WH3/ : CAPACITY;
  CUSTOMER / C1, C2, C3, C4/ : DEMAND;
  ROUTES( WAREHOUSE, CUSTOMER) : COST, VOLUME;
ENDSETS

! The objective;
[OBJ] min = @SUM( ROUTES: COST * VOLUME);

! The demand constraints;
@FOR( CUSTOMER( J): [DEM]
  @SUM( WAREHOUSE( I): VOLUME( I, J)) >=
    DEMAND( J));

! The supply constraints;
@FOR( WAREHOUSE( I): [SUP]
  @SUM( CUSTOMER( J): VOLUME( I, J)) <=
    CAPACITY( I));

! Here are the parameters;
DATA:
  CAPACITY = 30, 25, 21 ;
  DEMAND = 15, 17, 22, 12;
  COST = 6, 2, 6, 7,
        4, 9, 5, 3,
        8, 8, 1, 5;
ENDDATA
END|
Ready NUM MOD Ln 31, Col 4 4:48
```

11 يمكن إدخال (كتابة) صياغة المشكلة في هذه الشاشة أو قراءتها من ملف نص.

```
MODEL:
! A 3 Warehouse, 4 Customer
  Transportation Problem;
SETS:
  WAREHOUSE / WH1, WH2, WH3/ : CAPACITY;
  CUSTOMER / C1, C2, C3, C4/ : DEMAND;
  ROUTES( WAREHOUSE, CUSTOMER) : COST, VOLUME;
ENDSETS

! The objective;
[OBJ] min = @SUM( ROUTES: COST * VOLUME);

! The demand constraints;
@FOR( CUSTOMER( J): [DEM]
  @SUM( WAREHOUSE( I): VOLUME( I, J)) >=
    DEMAND( J));

! The supply constraints;
@FOR( WAREHOUSE( I): [SUP]
  @SUM( CUSTOMER( J): VOLUME( I, J)) <=
    CAPACITY( I));

! Here are the parameters;
DATA:
  CAPACITY = 30, 25, 21 ;
  DEMAND = 15, 17, 22, 12;
  COST = 6, 2, 6, 7,
        4, 9, 5, 3,
        8, 8, 1, 5;
ENDDATA
END
```

Display a picture of model matrix

NUM MOD Ln 31, Col 4 4:51

```
LINGO - LINGO Model - LINGO1
File Edit LINGO Window Help

LINGO Model - LINGO1 Solve

MODEL:
! A 3 Warehouse, 4 Customer
  Transportation Problem;
SETS:
  WAREHOUSE / WH1, WH2, WH3/ : CAPACITY;
  CUSTOMER / C1, C2, C3, C4/ : DEMAND;
  ROUTES( WAREHOUSE, CUSTOMER) : COST, VOLUME;
ENDSETS

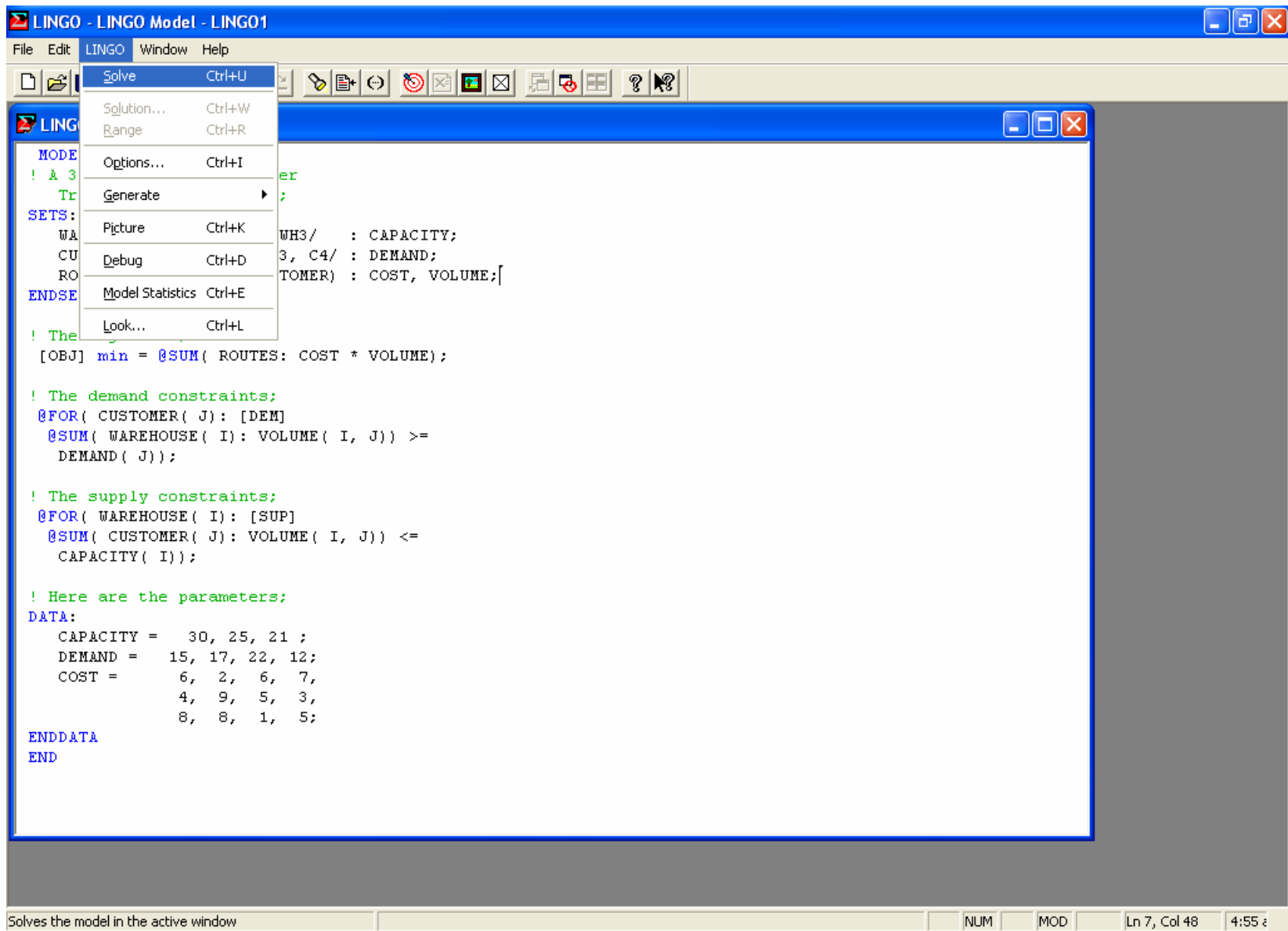
! The objective;
[OBJ] min = @SUM( ROUTES: COST * VOLUME);

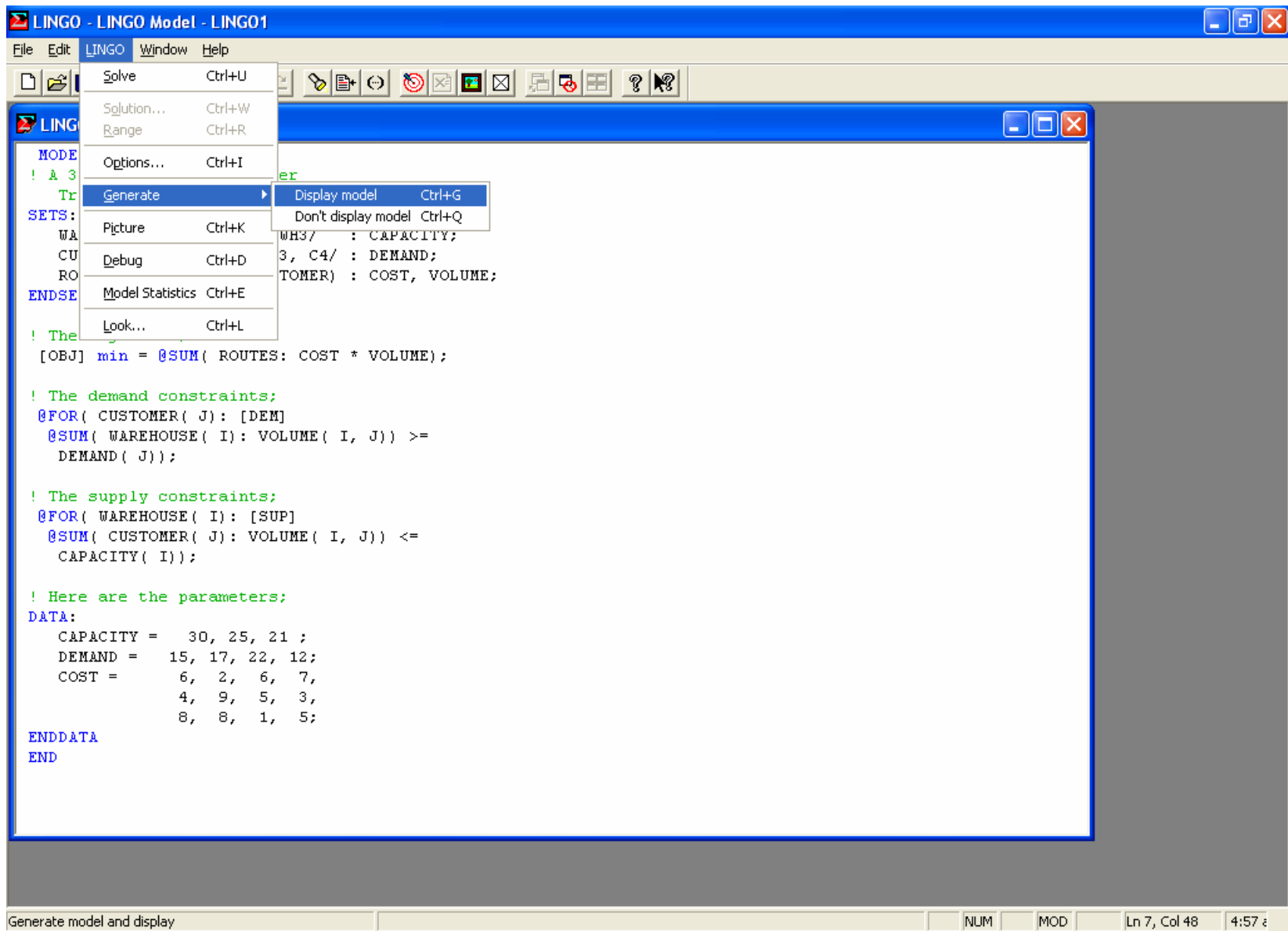
! The demand constraints;
@FOR( CUSTOMER( J): [DEM]
  @SUM( WAREHOUSE( I): VOLUME( I, J)) >=
  DEMAND( J));

! The supply constraints;
@FOR( WAREHOUSE( I): [SUP]
  @SUM( CUSTOMER( J): VOLUME( I, J)) <=
  CAPACITY( I));

! Here are the parameters;
DATA:
  CAPACITY = 30, 25, 21 ;
  DEMAND = 15, 17, 22, 12;
  COST = 6, 2, 6, 7,
        4, 9, 5, 3,
        8, 8, 1, 5;
ENDDATA
END

Solves the model in the active window
NUM MOD Ln 31, Col 4 4:52
```





```
LINGO - Generated Model Report - LINGO1
File Edit LINGO Window Help
[Icons]
LINGO Model - LINGO1
Generated Model Report - LINGO1
MODEL:
[OBJ] MIN= 6 * VOLUME_WH1_C1 + 2 * VOLUME_WH1_C2 + 6 * VOLUME_WH1_C3 + 7
* VOLUME_WH1_C4 + 4 * VOLUME_WH2_C1 + 9 * VOLUME_WH2_C2 + 5 *
VOLUME_WH2_C3 + 3 * VOLUME_WH2_C4 + 8 * VOLUME_WH3_C1 + 8 *
VOLUME_WH3_C2 + VOLUME_WH3_C3 + 5 * VOLUME_WH3_C4 ;
[DEM_C1] VOLUME_WH1_C1 + VOLUME_WH2_C1 + VOLUME_WH3_C1 >= 15 ;
[DEM_C2] VOLUME_WH1_C2 + VOLUME_WH2_C2 + VOLUME_WH3_C2 >= 17 ;
[DEM_C3] VOLUME_WH1_C3 + VOLUME_WH2_C3 + VOLUME_WH3_C3 >= 22 ;
[DEM_C4] VOLUME_WH1_C4 + VOLUME_WH2_C4 + VOLUME_WH3_C4 >= 12 ;
[SUP_WH1] VOLUME_WH1_C1 + VOLUME_WH1_C2 + VOLUME_WH1_C3 + VOLUME_WH1_C4
<= 30 ;
[SUP_WH2] VOLUME_WH2_C1 + VOLUME_WH2_C2 + VOLUME_WH2_C3 + VOLUME_WH2_C4
<= 25 ;
[SUP_WH3] VOLUME_WH3_C1 + VOLUME_WH3_C2 + VOLUME_WH3_C3 + VOLUME_WH3_C4
<= 21 ;
END
4, 9, 5, 3,
8, 8, 1, 5;
ENDDATA
END
For Help, press F1
NUM Ln 1, Col 1 4:59
```


LINGO - Solution Report - LINGO1

File Edit LINGO Window Help

LINGO Model - LINGO1

```

MODEL:
! A 3 Warehouse, 4 Customer
SETS
W
C
P
ENDS
! TH
[OE
! TH
@FC
@S
I
! TH
@FC
@S
C
! He
DATA
C
I
C
ENDDATA
END

```

Solution Report - LINGO1

Global optimal solution found.
Objective value: 161.0000
Total solver iterations: 6

Variable	Value	Reduced Cost
CAPACITY(WH1)	30.00000	0.000000
CAPACITY(WH2)	25.00000	0.000000
CAPACITY(WH3)	21.00000	0.000000
DEMAND(C1)	15.00000	0.000000
DEMAND(C2)	17.00000	0.000000
DEMAND(C3)	22.00000	0.000000
DEMAND(C4)	12.00000	0.000000
COST(WH1, C1)	6.000000	0.000000
COST(WH1, C2)	2.000000	0.000000
COST(WH1, C3)	6.000000	0.000000
COST(WH1, C4)	7.000000	0.000000
COST(WH2, C1)	4.000000	0.000000
COST(WH2, C2)	9.000000	0.000000
COST(WH2, C3)	5.000000	0.000000
COST(WH2, C4)	3.000000	0.000000
COST(WH3, C1)	8.000000	0.000000
COST(WH3, C2)	8.000000	0.000000
COST(WH3, C3)	1.000000	0.000000
COST(WH3, C4)	5.000000	0.000000

Ready NUM Ln 1, Col 1 5:03

LINGO - Solution Report - LINGO1

File Edit LINGO Window Help

LINGO Model - LINGO1

MODEL:

```

! A 3-Warehouse, 4-Customer
SETS
  W / WH1, WH2, WH3 /;
  C / C1, C2, C3, C4 /;
  COST(W, C);
  VOLUME(W, C);
ENDSETS
! TH
[OBJ
! TH
@FC
@S
I
! TH
@FC
@S
! He
DATA
C
I
C
ENDDATA
END

```

Solution Report - LINGO1

	COST(WH3, C4)	5.000000	0.000000
	VOLUME(WH1, C1)	2.000000	0.000000
	VOLUME(WH1, C2)	17.000000	0.000000
	VOLUME(WH1, C3)	1.000000	0.000000
	VOLUME(WH1, C4)	0.000000	2.000000
	VOLUME(WH2, C1)	13.000000	0.000000
	VOLUME(WH2, C2)	0.000000	9.000000
	VOLUME(WH2, C3)	0.000000	1.000000
	VOLUME(WH2, C4)	12.000000	0.000000
	VOLUME(WH3, C1)	0.000000	7.000000
	VOLUME(WH3, C2)	0.000000	11.000000
	VOLUME(WH3, C3)	21.000000	0.000000
	VOLUME(WH3, C4)	0.000000	5.000000
	Row	Slack or Surplus	Dual Price
	OBJ	161.0000	-1.000000
	DEM(C1)	0.000000	-6.000000
	DEM(C2)	0.000000	-2.000000
	DEM(C3)	0.000000	-6.000000
	DEM(C4)	0.000000	-5.000000
	SUP(WH1)	10.000000	0.000000
	SUP(WH2)	0.000000	2.000000
	SUP(WH3)	0.000000	5.000000

Ready

NUM

Ln 1, Col 1

5:04

The screenshot shows the LINGO software interface. The title bar reads "LINGO - LINGO Model - LINGO1". The menu bar includes "File", "Edit", "LINGO", "Window", and "Help". The "LINGO" menu is open, showing options: "Solve" (Ctrl+U), "Solution..." (Ctrl+W), "Range" (Ctrl+R), "Options..." (Ctrl+I), "Generate" (with a right-pointing arrow), "Picture" (Ctrl+K), "Debug" (Ctrl+D), "Model Statistics" (Ctrl+E), and "Look..." (Ctrl+L). The main window contains the following LINGO code:

```

MODEL
! A 3
Tr
SETS:
WA
CU
RO
ENDSET

! The objective;
[OBJ] min = @SUM( ROUTES: COST * VOLUME);

! The demand constraints;
@FOR( CUSTOMER( J): [DEM]
@SUM( WAREHOUSE( I): VOLUME( I, J)) >=
DEMAND( J));

! The supply constraints;
@FOR( WAREHOUSE( I): [SUP]
@SUM( CUSTOMER( J): VOLUME( I, J)) <=
CAPACITY( I));

! Here are the parameters;
DATA:
CAPACITY = 30, 25, 21 ;
DEMAND = 15, 17, 22, 12;
COST = 6, 2, 6, 7,
4, 9, 5, 3,
8, 8, 1, 5;
ENDDATA
END

```

At the bottom of the window, a status bar displays: "Create a range report for this model" | NUM | MOD | Ln 1, Col 1 | 5:10

LINGO - Range Report - LINGO1

File Edit LINGO Window Help

LINGO Model - LINGO1

MODEL:

```

! A 3 Ware
  Transpo
SETS:
  WAREHOUS
  CUSTOMER
  ROUTES (
ENDSETS
! The obje
[OBJ] min
! The dema
@FOR( CUS
@SUM( WA
  DEMAND (
! The supp
@FOR( WAR
@SUM( CU
  CAPACIT
! Here are
DATA:
  CAPACIT
  DEMAND =
  COST =
ENDDATA
END

```

Range Report - LINGO1

Ranges in which the basis is unchanged:

Objective Coefficient Ranges				
Variable	Current Coefficient	Allowable Increase	Allowable Decrease	
VOLUME (WH1, C1)	6.000000	2.000000	1.000000	
VOLUME (WH1, C2)	2.000000	9.000000	2.000000	
VOLUME (WH1, C3)	6.000000	1.000000	5.000000	
VOLUME (WH1, C4)	7.000000	INFINITY	2.000000	
VOLUME (WH2, C1)	4.000000	1.000000	2.000000	
VOLUME (WH2, C2)	9.000000	INFINITY	9.000000	
VOLUME (WH2, C3)	5.000000	INFINITY	1.000000	
VOLUME (WH2, C4)	3.000000	2.000000	5.000000	
VOLUME (WH3, C1)	8.000000	INFINITY	7.000000	
VOLUME (WH3, C2)	8.000000	INFINITY	11.00000	
VOLUME (WH3, C3)	1.000000	5.000000	INFINITY	
VOLUME (WH3, C4)	5.000000	INFINITY	5.000000	

Righthand Side Ranges				
Row	Current RHS	Allowable Increase	Allowable Decrease	
DEM (C1)	15.00000	10.00000	2.000000	
DEM (C2)	17.00000	10.00000	17.00000	
DEM (C3)	22.00000	10.00000	1.000000	
DEM (C4)	12.00000	10.00000	2.000000	
SUP (WH1)	30.00000	INFINITY	10.00000	
SUP (WH2)	25.00000	2.000000	10.00000	
SUP (WH3)	21.00000	1.000000	10.00000	

For Help, press F1

NUM

Ln 1, Col 1

5:12

الواجب

- على الطالب تطبيق هذا المثال باستخدام أسلوب Lindo مرة وبأسلوب Lingo مرة أخرى.
- إعادة حل مشكلة النقل في الدروس السابقة بالطريقتين.

مع تمنياتي بالتوفيق
وكل عام وأنتم بخير،،،