Exercise 3-4-4. Read the following statements carefully to see whether it is *true* or *false*. Justify your answer by constructing a simple illustrative example (for false), if possible, or by a simple proof (if true).

1. If an LP has more than one optimal solution, it must have an infinite number of optimal solutions.

2. The unboundedness criterion will never be satisfied while solving Phase I of the two–phase simplex method.

3. In solving an LP by the simplex method, a new feasible solution is generated after every pivot step.

Although we require the modified rules to guarantee finite termination of the simplex method, it has been observed in practice that cycling is a very rare phenomenon (Kotiah & Steinberg 1977, Kotiah & Steinberg 1978). Thus anti–cycling rules are not generally used, since they tend to increase the number of pivots required (Gass 1985, p. 183). A frequently used practical technique is to turn on anti-cycling rules if the simplex method "stalls" at the same vertex for a large number of pivots. Other well known techniques are Devex (Harris 1973) and Expand (Gill, Murray, Saunders & Wright 1989).

3.5 Reduction to Standard Form

We now know from Theorem 3.4.2, that the two–phase simplex method can process any linear program in standard form. In this section, we show that any linear program can be rewritten in standard form. Hence the simplex method can process any linear program whatever.

We first note that

$$\max_{x \in S} f(x) = -\min_{x \in S} -f(x)$$

for any function f and any set S, since

$$\bar{x} \in S, f(\bar{x}) = \max_{x \in S} f(x) \iff \bar{x} \in S \text{ and } f(\bar{x}) \geq f(x), \text{ for all } x \in S$$

$$\iff \bar{x} \in S \text{ and } -f(\bar{x}) \leq -f(x), \text{ for all } x \in S$$

$$\iff -f(\bar{x}) = \min_{x \in S} -f(x), \bar{x} \in S$$

$$\iff f(\bar{x}) = -\min_{x \in S} -f(x), \bar{x} \in S$$

Consequently, to solve a linear program $\max_{x \in S} p'x$, we solve $\min_{x \in S} -p'x$ and report the negative of the minimum value as the optimal value of the linear program.

Furthermore, if we are given less-than inequalities, multiplication of these inequalities by -1 results in greater-than inequalities.

Example 3-5-1.

$$
\begin{array}{rrcl}
\text{max} & x_1 + x_2 - x_3 & & \\
\text{subject to} & x_1 + x_2 + x_3 & \leq & 4 \\
& 2x_1 - x_2 & \geq & 2 \\
& -x_1 - x_2 + x_3 & \leq & 10 \\
& x_1 + 2x_2 - x_3 & \geq & 1 \\
& x_1, x_2, x_3 & \geq & 0
\end{array}
$$

```
≫ load ex3-5-1
≫ A(1,:)  = -A(1,:); b(1) = -b(1);
≫ A(3,:)  = -A(3,:); b(3) = -b(3);
≫ T = totbl(A,b,-p);
```

		x_1	x_2	x_3	1
x_4	=	-1	-1	-1	4
x_5	=	2	-1	0	-2
x_6	=	1	1	-1	10
x_7	=	1	2	-1	-1
z	=	-1	-1	1	0

Since this tableau is in standard form, we can apply the two–phase simplex method to solve it. The details are left to the reader.

Exercise 3-5-2. Use the two–phase simplex procedure to solve

$$
\begin{array}{rrrrcl}
\text{max} & & & z = 2x_1 + 3x_2 & & \\
& -4x_1 & + & 3x_2 & \leq & 12 \\
& 2x_1 & + & x_2 & \leq & 6 \\
\text{subject to} & x_1 & + & x_2 & \geq & 3 \\
& 5x_1 & + & x_2 & \geq & 4 \\
& x_i & \geq & 0, & & i = 1,2
\end{array}
$$

Using the above simple reductions, we see that the most general linear program is of the form:

$$
\begin{array}{rrcl}
\min_{x,y} & p'x + q'y & & \\
\text{subject to} & Ax + By & \geq & a \\
& Cx + Dy & = & b \\
& x & \geq & 0
\end{array}
\tag{3.9}
$$

Note the presence of equalities and the free variables y, which for expositional purposes we have arranged to come after inequalities and nonnegative variables. A mechanical but somewhat inefficient way to reduce (3.9) to standard form is to replace the free variables by a difference between their positive and negative parts, and to replace equations by two inequalites as follows:

$$ y \text{ free} \iff y = y_+ - y_-, \qquad y_+, y_- \geq 0 $$

$$ Cx + Dy = b \iff \begin{array}{l} Cx + Dy \geq b \\ Cx + Dy \leq b \end{array} $$

Upon substituting for y and replacing the equalities by inequalities, problem (3.9) becomes a linear program in standard form

$$
\begin{array}{rl}
\min_{x,y_+,y_-} & p'x + q'y_+ - q'y_- \\
\text{subject to} & Ax + By_+ - By_- \geq a \\
& Cx + Dy_+ - Dy_- \geq b \\
& -Cx - Dy_+ + Dy_- \geq -b \\
& x, y_+, y_- \geq 0
\end{array}
\tag{3.10}
$$

By collecting the variables x, y_+ and y_- into a vector t and defining H, h and c by

$$
H = \begin{bmatrix} A & B & -B \\ C & D & -D \\ -C & -D & D \end{bmatrix}, \quad
h = \begin{bmatrix} a \\ b \\ -b \end{bmatrix}, \quad
c = \begin{bmatrix} p \\ q \\ -q \end{bmatrix}
$$

we see that the above problem can then be rewritten in standard form as

$$
\begin{array}{rl}
\min_t & c't \\
\text{subject to} & Ht \geq h \\
& t \geq 0
\end{array}
$$

We now show an example of how this process (which we call *Scheme I*) works.

Example 3-5-3.

$$
\begin{array}{rl}
\min & 2x_1 - x_2 + x_3 \\
\text{subject to} & x_1 - x_2 + x_3 \geq -1 \\
& x_1 - x_2 - x_3 \geq 2 \\
& x_1 + x_2 + x_3 = 3 \\
& x_1, x_2 \geq 0
\end{array}
$$

Note that x_3 is a free variable, and the third constraint is an equality. Therefore, we need to split x_3 into its positive and negative parts and replace the third equation by two inequalities. We do this by creating new matrices H, h and c as described above and then constructing the initial tableau:

```
>> A = [1 -1; 1 -1]; B = [1; -1];
>> C = [1 1]; D = 1;
>> a = [-1; 2]; b = 3;
>> p = [2; -1]; q = 1;
>> H = [A B -B; C D -D; -C -D D];
>> h = [a; b; -b];
>> c = [p; q; -q];
>> T = totbl(H,h,c,0,'t');
```

		t_1	t_2	t_3	t_4	1
t_5	=	1	−1	1	−1	1
t_6	=	1	−1	−1	1	−2
t_7	=	1	1	1	−1	−3
t_8	=	−1	−1	−1	1	3
z	=	2	−1	1	−1	0

We also relabel the variables t when we set up the tableau. This is avoid any confusion resulting from the fact that $x_3 = t_3 - t_4$. We now perform the Phase I method on this problem:

```
>> T = addcol(T,[0 1 1 0 0]','x0',5);
>> w = [0 0 0 0 1 0];
>> T = addrow(T,w,'z0',6);
```

		t_1	t_2	t_3	t_4	x_0	1
t_5	=	1	−1	1	−1	0	1
t_6	=	1	−1	−1	1	1	−2
t_7	=	1	1	1	−1	1	−3
t_8	=	−1	−1	−1	1	0	3
z	=	2	−1	1	−1	0	0
z_0	=	0	0	0	0	1	0

The next step is the special pivot that generates a feasible tableau for Phase I.

```
>> T = ljx(T,3,5);
```

		t_1	t_2	t_3	t_4	t_7	1
t_5	=	1	−1	1	−1	0	1
t_6	=	0	−2	−2	2	1	1
x_0	=	−1	−1	−1	1	1	3
t_8	=	−1	−1	−1	1	0	3
z	=	2	−1	1	−1	0	0
z_0	=	−1	−1	−1	1	1	3

A further pivot leads to an optimal Phase I tableau.

$\gg$ T = ljx(T,3,1);

		x_0	t_2	t_3	t_4	t_7	1
t_5	=	-1	-2	0	0	1	4
t_6	=	0	-2	-2	2	1	1
t_1	=	-1	-1	-1	1	1	3
t_8	=	1	0	0	0	-1	0
z	=	-2	-3	-1	1	2	6
z_0	=	1	0	0	0	0	0

Striking out the x_0 column and the z_0 row, we note that the resulting tableau is feasible, but not optimal. However, the following two pivots lead to an optimal tableau.

$\gg$ T = delcol(T,'x0');

$\gg$ T = delrow(T,'z0');

$\gg$ T = ljx(T,2,1);

$\gg$ T = ljx(T,1,3);

		t_6	t_3	t_5	t_7	1
t_4	=	0.5	1	-0.5	0	1.5
t_2	=	0	0	-0.5	0.5	2
t_1	=	0.5	0	0	0.5	2.5
t_8	=	0	0	0	-1	0
z	=	0.5	0	1	0.5	1.5

We now need to convert the optimal values for t back into optimal values for x. Note that $x_1 = t_1 = 2.5$, $x_2 = t_2 = 2$, $x_3 = t_3 - t_4 = -1.5$ and hence the optimal solution is $z = 1.5$.

The following theorem is now immediate.

Theorem 3.5.1. *Given any linear program, the two–phase simplex method with the smallest subscript anticycling rule terminates after a finite number of pivots with a conclusion that the problem is* infeasible, *or at an* optimal *or* unbounded *tableau.*

Exercise 3-5-4. Convert the following problem into standard form using Scheme I and solve using the simplex method:

$$
\begin{array}{llrcrcrcl}
\text{min} & & & & x + 2y + 3z & & & & \\
& & x & - & y & + & 3z & \leq & 3 \\
\text{subject to} & 4x & + & y & & & & \geq & 1 \\
& & & & & & z & \geq & 0
\end{array}
$$

Exercise 3-5-5. Convert the following problem to standard form and solve using the two-phase simplex method:

$$
\begin{array}{ll}
\max & -2x_1 - x_2 - x_3 - 2x_4 \\
\text{subject to} & x_1 - x_2 + x_3 - x_4 = -1 \\
& -x_1 - x_2 - x_3 - x_4 = -3 \\
& -x_1 + x_2 - x_3 + x_4 \le 1 \\
& x_1 + x_2 - x_3 - x_4 \le -1 \\
& -x_1 - x_2 + x_3 + x_4 \le 1 \\
& -x_1 + x_2 + x_3 - x_4 \le -2 \\
& x_1, x_4 \ge 0
\end{array}
$$

Note that the variables x_2, x_3 are free.

In Example 3-5-3, we only added one equation and one variable to the problem at hand. When there are multiple free variables and multiple equations, we end up adding many more equations and variables to the problem as is the case in Exercise 3-5-5. However, there is another technique that generates a standard form problem by just adding one extra variable and one extra constraint. It derives from the following observations:

$$
y \text{ free} \iff y = \hat{y} - e\eta, \qquad \hat{y} \ge 0, \qquad \eta \ge 0
$$

$$
Cx + Dy = b \iff \begin{array}{l} Cx + Dy \ge b \\ e'(Cx + Dy - b) \le 0 \end{array}
$$

Here e is a vector of ones of appropriate dimension. Upon substituting for y and replacing the equalities by inequalities, problem (3.9) becomes a linear program in standard form

$$
\begin{array}{lrl}
\min_{x,\hat{y},\eta} & p'x + q'(\hat{y} - e\eta) & \\
\text{subject to} & Ax + B(\hat{y} - e\eta) & \ge & a \\
& Cx + D(\hat{y} - e\eta) & \ge & b \\
& -e'Cx - e'D(\hat{y} - e\eta) & \ge & -e'b \\
& x, \hat{y}, \eta & \ge & 0
\end{array}
$$

Exercise 3-5-6. Use the above approach to solve the problem given in Exercise 3-5-5.

Exercise 3-5-7. Solve the following problem:

$$\max \quad 2y_1 - y_3$$

$$\text{subject to} \quad \begin{array}{rcrcrcrcl}
5y_1 & - & 2y_2 & + & y_3 & - & y_4 & = & 36 \\
y_1 & & & + & y_3 & & & \geq & 4 \\
y_1 & + & 3y_2 & + & y_3 & & & \geq & 1
\end{array}$$

$$y_1 \leq 8, \quad y_2 \geq 0, \quad y_3 \leq 0, \quad y_4 \geq 0$$

Does the problem have a unique optimal solution? Justify.

Exercise 3-5-8. Use the simplex method to solve the following problem:

$$\max \quad 2x_1 + 5x_2 - x_3$$

$$\text{subject to} \quad \begin{array}{rcrcrcl}
5x_1 & & & + & x_3 & \geq & 1 \\
-2x_1 & & & + & x_3 & \leq & 22 \\
-4x_1 & + & x_2 & - & x_3 & \leq & -6
\end{array}$$

$$x_1 \leq 0 \qquad x_2 \geq 3 \qquad x_3 \geq 0$$

Since the above approaches increase the number of variables and constraints, a more efficient way of reducing (3.9) to standard form is *Scheme II*.

The basic idea is simple and the changes relate only to the free variables and to equations. For the equations in the model, when we construct the initial tableau, a slack variable is associated with this constraint. Such slack variables must be zero in any solution, so we attempt to pivot all the slack variables x_{n+i} corresponding to the equations to the top of the tableau. Once they are at the top of the tableau, we are able to assign them the value 0. Such variables will never be chosen as pivot columns, so we can simply delete the corresponding column.

We pivot all the free variables $x_{N(j)}$ to the side of the tableau. Once they are on the side of the tableau, it does not matter what values we assign to the independent (nonbasic) variables, since we can always compute the values of the free variables using this defining equation. We therefore (implicitly) delete (and save for later computation) the defining equations corresponding to these free variables.

Example 3-5-9. We illustrate this approach on Example 3-5-3.

```
≫ load ex3-5-3
≫ T = totbl(A,b,p);
```

		x_1	x_2	x_3	1
x_4	=	1	-1	1	1
x_5	=	1	-1	-1	-2
x_6	=	1	1	1	-3
z	=	2	-1	1	0

We now remove the slack variable x_6 that corresponds to the equation using a Jordan exchange to move it to the top, followed by a deletion of that column. The following MATLAB operations effect this.

```
>> T = ljx(T,3,1);
>> T = delcol(T,'x6');
```

		x_2	x_3	1
x_4	=	-2	0	4
x_5	=	-2	-2	1
x_1	=	-1	-1	3
z	=	-3	-1	6

We now attempt to move the free variable x_3 to the side of the tableau. The following pivot carries this out.

```
>> T = ljx(T,2,2);
```

		x_2	x_5	1
x_4	=	-2	0	4
x_3	=	-1	-0.5	0.5
x_1	=	0	0.5	2.5
z	=	-2	0.5	5.5

Note that since x_3 is free, the second row of the tableau can be viewed as a defining relationship for x_3. We can solve the problem without x_3 and at the end use the optimal values of x_2 and x_5 to generate the optimal value for x_3. What we actually do is to move the corresponding row below the objective row so that we can easily ignore it through the rest of the tableau calculations. The routine `permrows` carries this out by permuting the rows of the given tableau according to the permutation specified as the second argument.

```
>> T = permrows(T,[1 3 4 2]);
```

		x_2	x_5	1
x_4	=	-2	0	4
x_1	=	0	0.5	2.5
z	=	-2	0.5	5.5
x_3	=	-1	-0.5	0.5

At this stage, all the variables left in the tableau x_2, x_5, x_4 and x_1 are nonnegative, and we are fortunate that this tableau is feasible. Hence, we can proceed directly with Phase II of the simplex method.

```
≫ T = ljx(T,1,1);
```

		x_4	x_5	1
x_2	=	-0.5	0	2
x_1	=	0	0.5	2.5
z	=	1	0.5	1.5
x_3	=	0.5	-0.5	-1.5

Note that this tableau is optimal, with $x_1 = 2.5$, $x_2 = 2$, and $x_3 = -1.5$. The optimal value is $z = 1.5$ as we had calculated in Example 3-5-3. Note that the final value of x_3 was read off the final tableau. Instead we could have substituted the optimal values of x_2 and x_5 into the penultimate tableau and recovered $x_3 = -x_2 - 0.5x_5 + 0.5 = -1.5$.

A final comment. The order in which we bring the slack variables associated with constraints to the top of the tableau and the free variables to the side is arbitrary. In fact we could solve the above problem using the following sequence of operations:

```
≫ load ex3-5-3
≫ T = totbl(A,b,p);
≫ T = ljx(T,3,3);
≫ T = delcol(T,'x6');
≫ T = permrows(T,[1 2 4 3]);
```

		x_1	x_2	1
x_4	=	0	-2	4
x_5	=	2	0	-5
z	=	1	-2	3
x_3	=	-1	-1	3

Note that here in one pivot we both moved the slack variable to the top and deleted it, and also moved the free variable to the side and removed it. It is usually advantageous to do this, but in this case the resulting tableau is not feasible. Hence we need to carry out a Phase I process to generate a feasible tableau.

The following MATLAB steps summarize this process. The first two steps create the Phase I problem in the tableau T.

```
≫ T = addcol(T,[0 1 0 0]','x0',3);
≫ T = addrow(T,[0 0 1 0],'z0',5);
```

		x_1	x_2	x_0	1
x_4	=	0	-2	0	4
x_5	=	2	0	1	-5
z	=	1	-2	0	3
x_3	=	-1	-1	0	3
z_0	=	0	0	1	0

The following sequence of pivots precisely carry out the standard Phase I procedure.

```
≫ T = ljx(T,2,3);
≫ T = ljx(T,2,1);
≫ T = delrow(T,'z0');
≫ T = delcol(T,'x0');
```

		x_2	x_5	1
x_4	=	-2	0	4
x_1	=	0	0.5	2.5
z	=	-2	0.5	5.5
x_3	=	-1	-0.5	0.5

Note that the last MATLAB operations above just remove the row and the column corresponding to the Phase I variables. A single Phase II pivot `ljx(T,1,1)` leads to the same optimal tableau that was given above.

Exercise 3-5-10. Carry out Scheme II on the problem given in Exercise 3-5-5. Ensure your solutions agree!

Note that it is possible for this scheme to run into difficulties with zero pivots. The following description shows how to overcome this.

(i) Free variables: Pivot all free variables $x_{\mathrm{N}(j)}$ to the side of the tableau, delete and save their transformed rows. If a free variable $x_{\mathrm{N}(j)}$ cannot be pivoted to the side, then either:

   (a) The problem is independent of $x_{\mathrm{N}(j)}$ if $c_j = 0$, so strike out column j.

   (b) The problem is unbounded if $c_j \neq 0$ and tableau is feasible.

(ii) Equations: Pivot all the slack variables x_{n+i} corresponding to the equations to the top of the tableau and delete their corresponding columns. If an x_{n+i} corresponding to an equality cannot be pivoted to the top, then this equality is linearly dependent and either:

   (a) Equality $x_{n+i} = 0$ is consistent, delete.

   (b) Equality $x_{n+i} = 0$ is inconsistent, problem is infeasible.

We now give some simple examples that display this approach.

Example 3-5-11. In the following examples we will assume that $x_1 \geq 0$, x_2 is a free variable, x_3 is a slack associated with an inequality and x_4 is an artificial slack associated with an equality.

```
≫ load ex3-5-11-1
≫ T = totbl(A,b,p);
```

		x_1	x_2	1
x_3	=	1	0	6
x_4	=	-1	0	4
z	=	2	-1	0

In this case, the free variable x_2 cannot be moved to the side of the tableau. Furthermore, the corresponding $c_2 = -1$ so that the problem is unbounded. Note that the linear program

$$
\begin{aligned}
\min_{x_1, x_2} \quad & 2x_1 - x_2 \\
\text{subject to} \quad & x_1 \geq -6 \\
& -x_1 = -4 \\
& x_1 \geq 0
\end{aligned}
$$

is clearly unbounded by setting $x_1 = 4$, and letting x_2 go to ∞. It is also clear that if $c_2 = 0$, then the objective function becomes $\min 2x_1$ and hence the problem is independent of x_2 which should therefore be discarded. If instead, $h_1 = -6$, a Phase I procedure should be carried out to determine that the problem is infeasible.

Now we consider the same example with slightly different matrices A, b and p.

```
≫ load ex3-5-11-2
≫ T = totbl(A,b,p);
```

		x_1	x_2	1
x_3	=	1	-2	2
x_4	=	0	0	4
z	=	2	-1	0

In this case, the artificial variable x_4 cannot be moved to the top of the tableau. Furthermore, the corresponding $h_2 = 4$ so that the problem is infeasible. Note that the corresponding linear program

$$
\begin{aligned}
\min_{x_1, x_2} \quad & 2x_1 - x_2 \\
\text{subject to} \quad & x_1 - 2x_2 \geq -2 \\
& 0 = -4 \\
& x_1 \geq 0
\end{aligned}
$$

is clearly infeasible. In the case that $h_2 = 0$, the second constraint would read $0 = 0$, and can be discarded without any difficulty.

```
≫ load ex3-5-11-2
≫ b(2) = 0; T = totbl(A,b,p);
≫ T = delrow(T,'x4');
```

		x_1	x_2	1
x_3	=	1	-2	2
z	=	2	-1	0

In this example, we then pivot the free variable x_2 to the side resulting in the following (optimal) tableau:

$\gg$ T = ljx(T,1,2);

$$
\begin{array}{c c}
 & \begin{array}{ccc} x_1 & x_3 & 1 \end{array} \\
x_2 \quad = & \boxed{\begin{array}{ccc} 0.5 & -0.5 & 1 \\ 1.5 & 0.5 & -1 \end{array}} \\
z \quad = &
\end{array}
$$

Exercise 3-5-12. Consider the linear program:

$$
\begin{array}{ll}
\min & -x_1 + x_2 - x_3 - 4x_4 \\
\text{subject to} & x_1 - x_2 + 2x_3 - x_4 \;=\; -4 \\
& -x_1 + x_2 - x_3 - x_4 \;=\; 4 \\
& x_1 + x_2 - x_3 - x_4 \;\geq\; 2 \\
& -x_1 - x_2 + x_3 - 2x_4 \;\geq\; -4 \\
& x_1 + x_2 + 2x_3 + x_4 \;\geq\; -3 \\
& x_1, x_2 \;\geq\; 0 \\
& x_3, x_4 \qquad \text{free}
\end{array}
$$

(a) Use Scheme I to reformulate and solve the above problem as a standard form linear program with 5 variables and 6 constraints.

(b) Use Scheme II to solve the original problem in MATLAB. First, eliminate the free variables and the slack variables associated with the equations by appropriate pivot operations, then set up the Phase I problem. Use Example 3-5-9 as a template to solve this problem, and if necessary the Phase II problem. Be sure to determine the solution of the original problem.

Exercise 3-5-13. Convert the following problem into standard form (using Scheme II) and solve:

$$
\begin{array}{ll}
\max & x_1 + 4x_2 + x_3 \\
& 2x_1 + 2x_2 + x_3 \;=\; 4 \\
\text{subject to} & x_1 \qquad\quad - x_3 \;=\; 1 \\
& x_2 \;\geq\; 0, \quad x_3 \;\geq\; 0
\end{array}
$$

Exercise 3-5-14. Solve the following problem:

$$
\begin{array}{ll}
\min & z = -3x_1 + x_2 + 3x_3 - x_4 \\
& x_1 + 2x_2 - x_3 + x_4 \;=\; 0 \\
\text{subject to} & 2x_1 - 2x_2 + 3x_3 + 3x_4 \;=\; 9 \\
& x_1 - x_2 + 2x_3 - x_4 \;=\; 6 \\
& x_i \;\geq\; 0, \qquad i = 1, 2, 3, 4
\end{array}
$$

Exercise 3-5-15. Solve the following problem in MATLAB.

$$\text{min} \qquad c'x$$
$$\text{subject to} \quad Ax = b, \quad x \geq 0$$

where

$$A = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 3 & -1 & 2 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 7 \\ 7 \\ 5 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 6 \\ 1 \\ 1 \end{bmatrix}$$

3.6   Canonical Form and Tableau

The *long tableau* is a rewriting of the canonical form (3.2) as follows:

x_{B}	x_{N}	z	$= 1$
I	$-A$	0	$-b$
0	$-p'$	1	0

This is tableau notation for

$$x_{\text{B}} - Ax_{\text{N}} = -b, \qquad -p'x_{\text{N}} + z = 0$$

A pivot in this tableau corresponds again to interchanging a basic variable (now identified by a column of the identity matrix and zero in the bottom row) and a nonbasic variable. The pivoting rules are such that the entries appearing in the long tableau in the nonbasic columns are the negative of those in the condensed tableau and the last column is identical to that of the condensed tableau. Since the long tableau is just an alternate (and somewhat wasteful) method of bookkeeping, we shall not dwell on it further in this book.