

# 3장 MATLAB 프로그래밍

3.1 M-파일

3.2 입력-출력

3.3 구조 프로그래밍

3.4 내포화와 들여쓰기

3.5 M-파일로의 함수 전달

3.6 사례연구: 번지점프하는 사람의 속도



# 3장 MATLAB 프로그래밍

## ■ 수학적 모델:

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

## ■ Euler법:

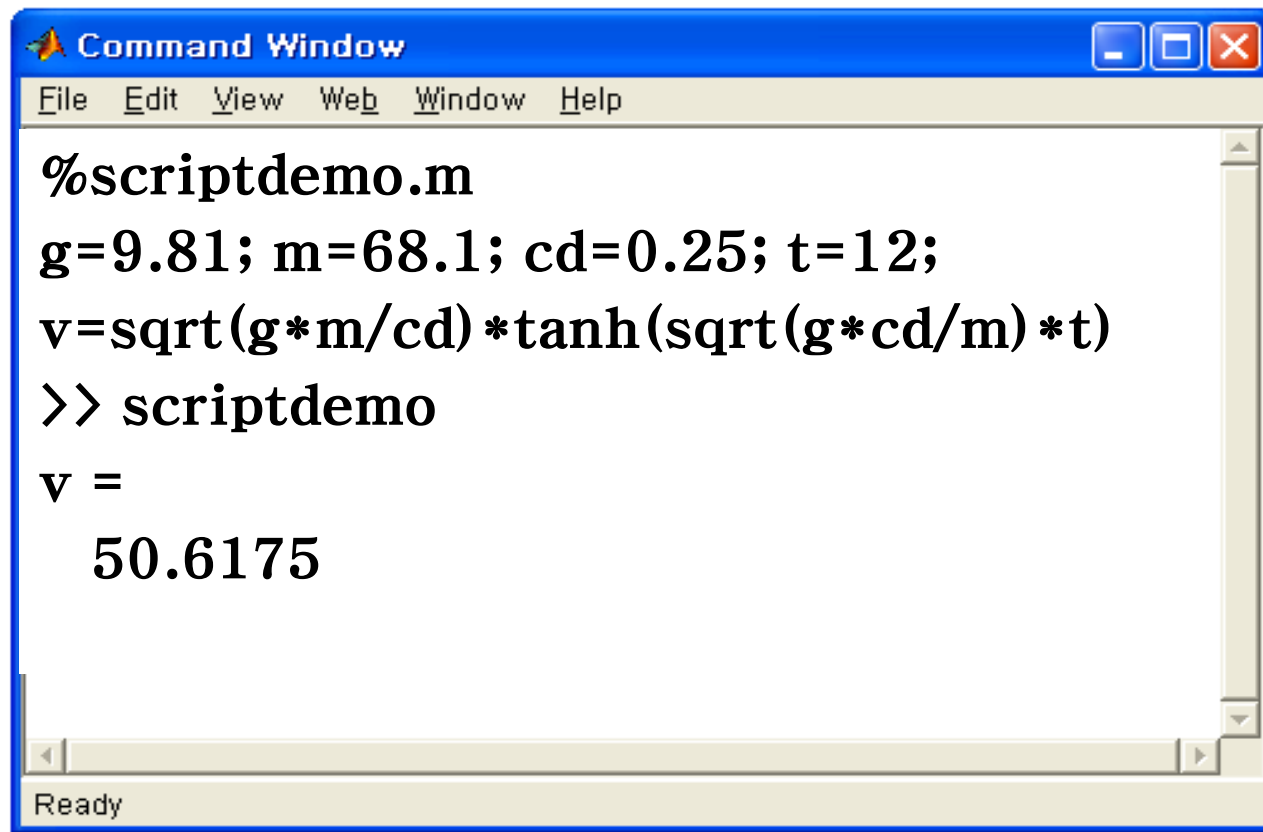
$$v_{i+1} = v_i + \frac{dv_i}{dt} \Delta t$$



## 3.1 M-파일 (1/5)

### ■ 스크립트 파일

- 일련의 MATLAB 명령어를 구성되어 저장된 M-파일이다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, Help. The main area of the window contains MATLAB code and its output. The code is: `%scriptdemo.m`, `g=9.81; m=68.1; cd=0.25; t=12;`, `v=sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t)`, and `>> scriptdemo`. The output is `v =` followed by `50.6175`. At the bottom of the window, there is a status bar that says "Ready".

```
Command Window
File Edit View Web Window Help

%scriptdemo.m
g=9.81; m=68.1; cd=0.25; t=12;
v=sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t)
>> scriptdemo

v =
    50.6175

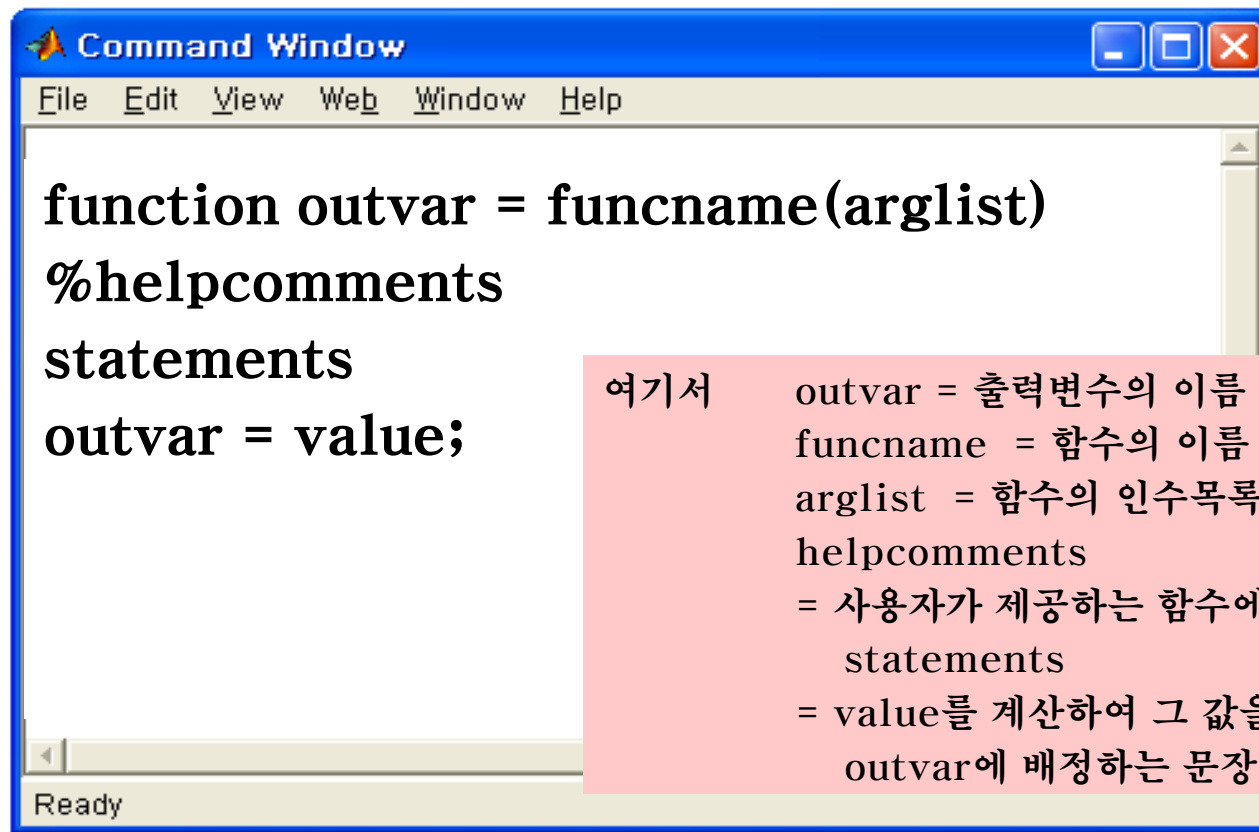
Ready
```



## 3.1 M-파일 (2/5)

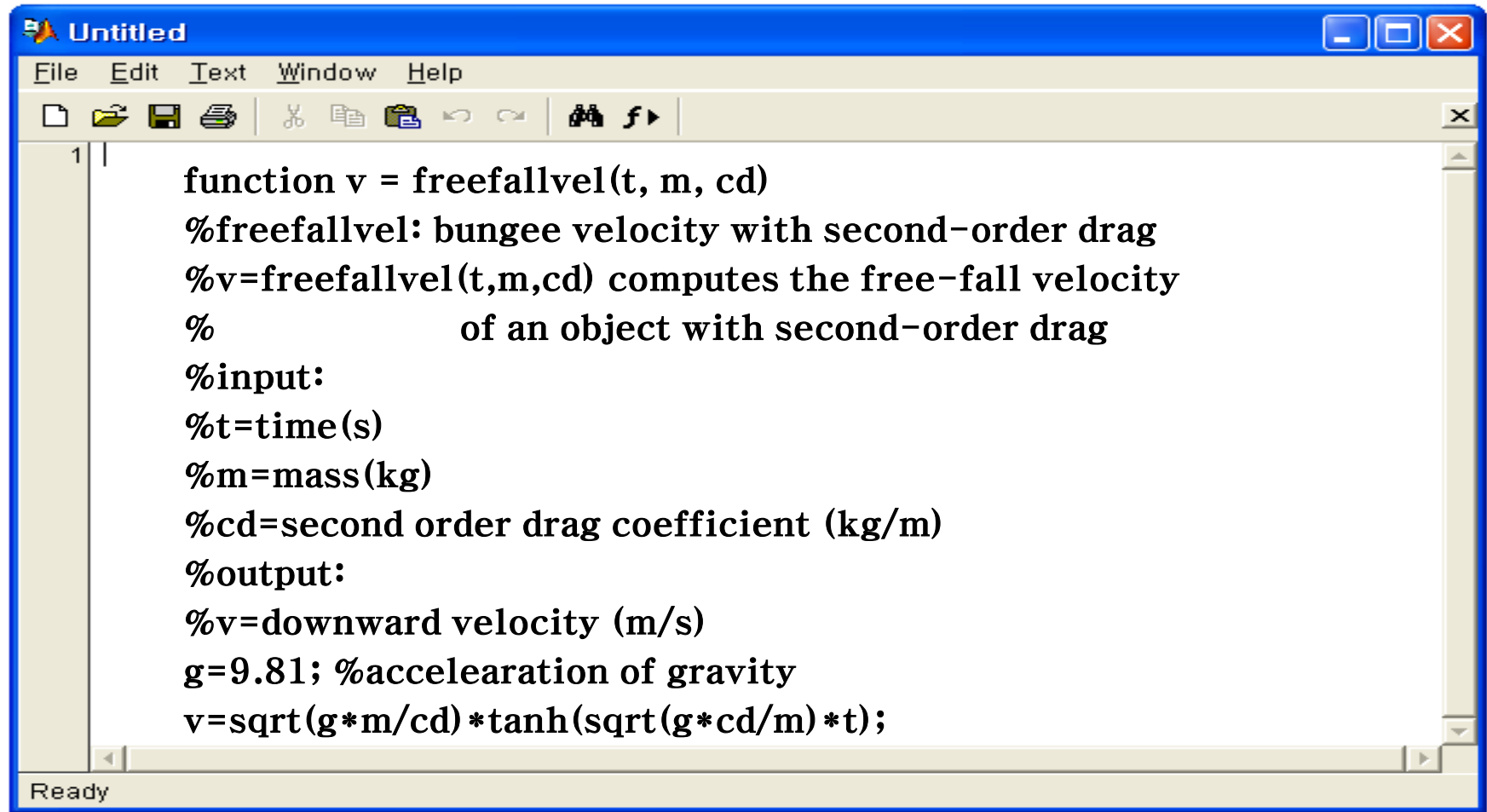
### ■ 함수 파일

- function이라는 단어로 시작하는 M-파일이다.



## 예제 3.2 [1/3]

Q. 번지점프하는 사람의 자유낙하 속도를 함수 파일을 사용하여 구하라.



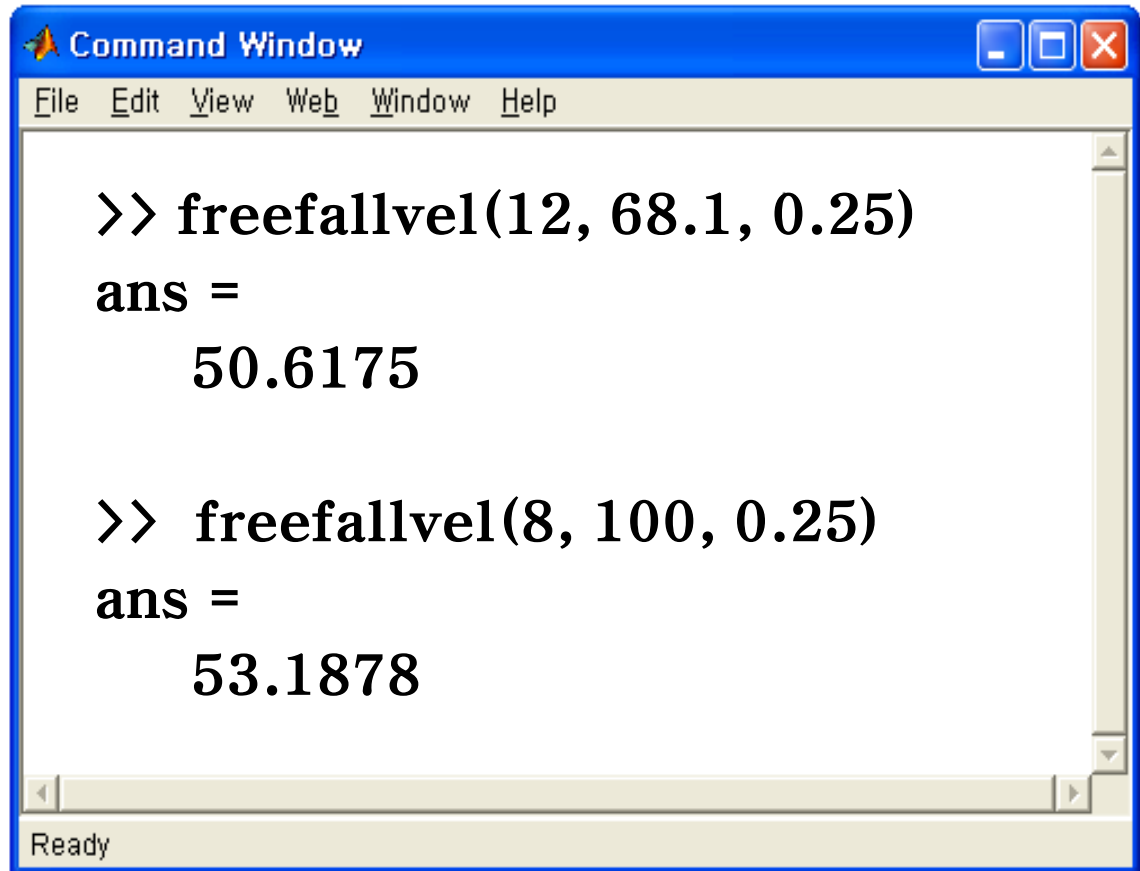
```
function v = freefallvel(t, m, cd)
%freefallvel: bungee velocity with second-order drag
%v=freefallvel(t,m,cd) computes the free-fall velocity
%               of an object with second-order drag
%input:
%t=time(s)
%m=mass(kg)
%cd=second order drag coefficient (kg/m)
%output:
%v=downward velocity (m/s)
g=9.81; %acceleartion of gravity
v=sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);
```



## 예제 3.2 [2/3]

Q1. 68.1 kg 인  
사람의 12초 후의  
속도를 구하려면...

Q2. 100 kg 인  
사람의 8초 후의  
속도를 구하려면...

A screenshot of the MATLAB Command Window. The title bar is blue with the text 'Command Window' and standard window control buttons (minimize, maximize, close). The menu bar includes 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main text area shows two MATLAB commands and their outputs. The first command is '>> freefallvel(12, 68.1, 0.25)' followed by 'ans = 50.6175'. The second command is '>> freefallvel(8, 100, 0.25)' followed by 'ans = 53.1878'. The status bar at the bottom says 'Ready'.

```
Command Window
File Edit View Web Window Help

>> freefallvel(12, 68.1, 0.25)
ans =
    50.6175

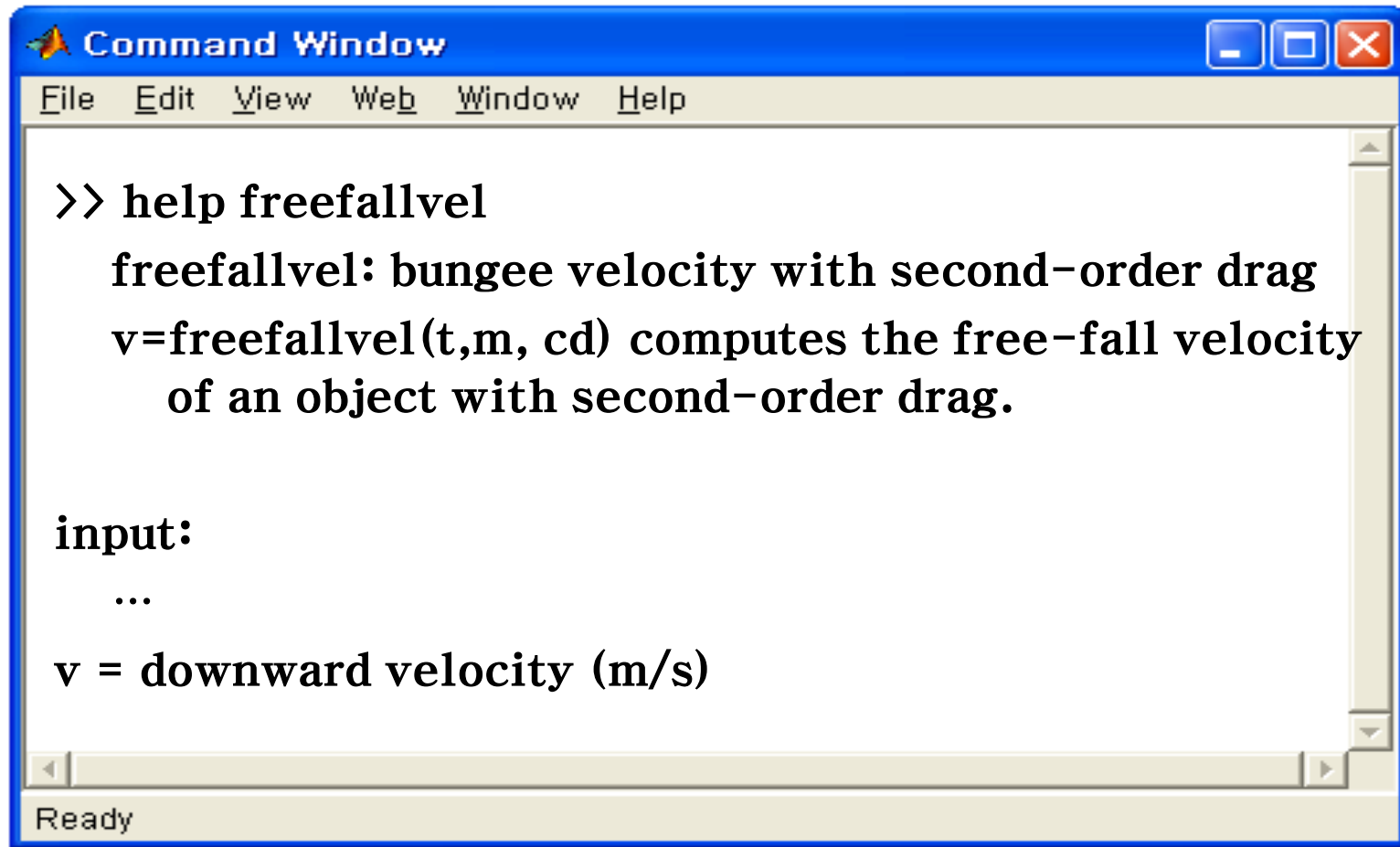
>> freefallvel(8, 100, 0.25)
ans =
    53.1878

Ready
```



## 예제 3.2 [3/3]

도움 설명을 불러내려면 다음과 같이 입력한다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window contains the following text:

```
>> help freefallvel  
freefallvel: bungee velocity with second-order drag  
v=freefallvel(t,m, cd) computes the free-fall velocity  
of an object with second-order drag.  
  
input:  
...  
v = downward velocity (m/s)
```

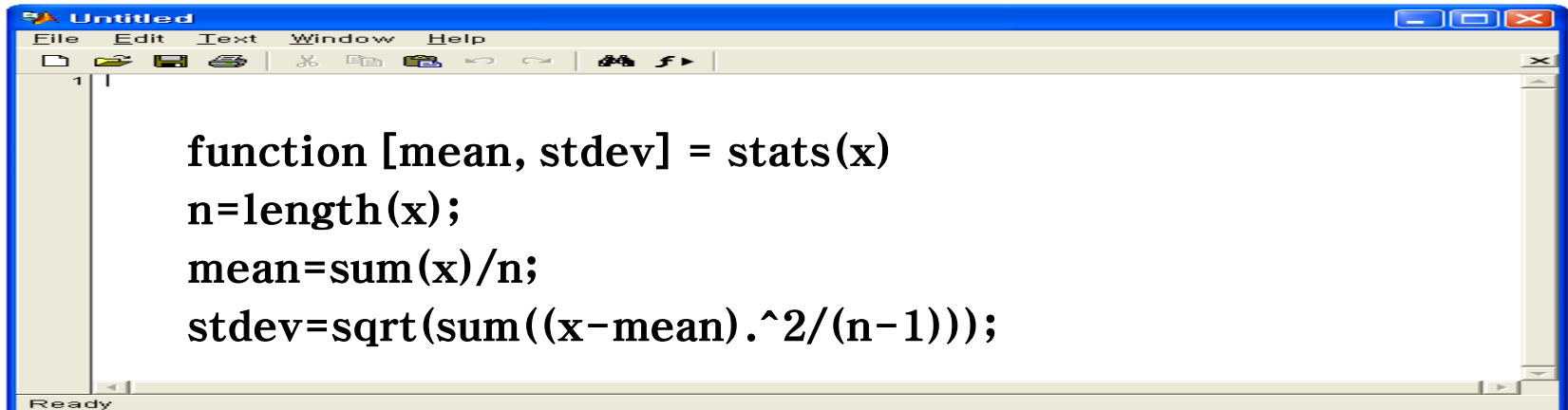
At the bottom of the window, there is a status bar that says "Ready".



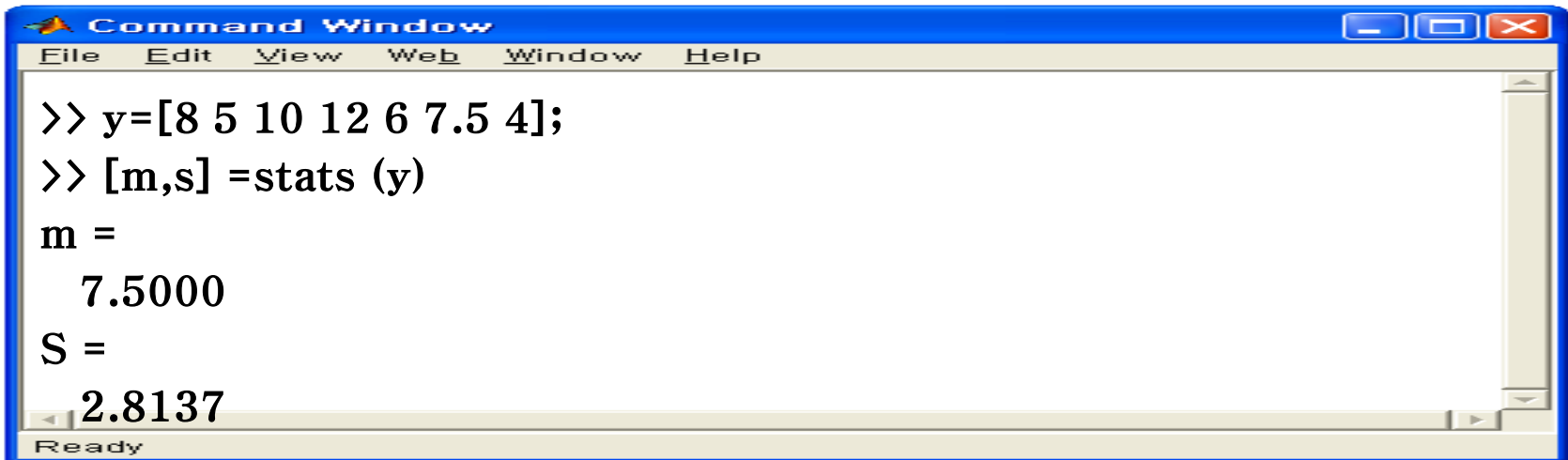
## 3.1 M-파일 (3/5)

- 함수 M-파일은 2개 이상의 결과를 반환할 수 있다.

예) 벡터의 평균과 표준편차의 계산



```
function [mean, stdev] = stats(x)
n=length(x);
mean=sum(x)/n;
stdev=sqrt(sum((x-mean).^2/(n-1)));
```



```
>> y=[8 5 10 12 6 7.5 4];
>> [m,s] =stats (y)
m =
    7.5000
S =
    2.8137
```

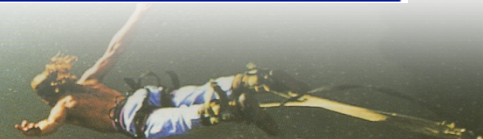
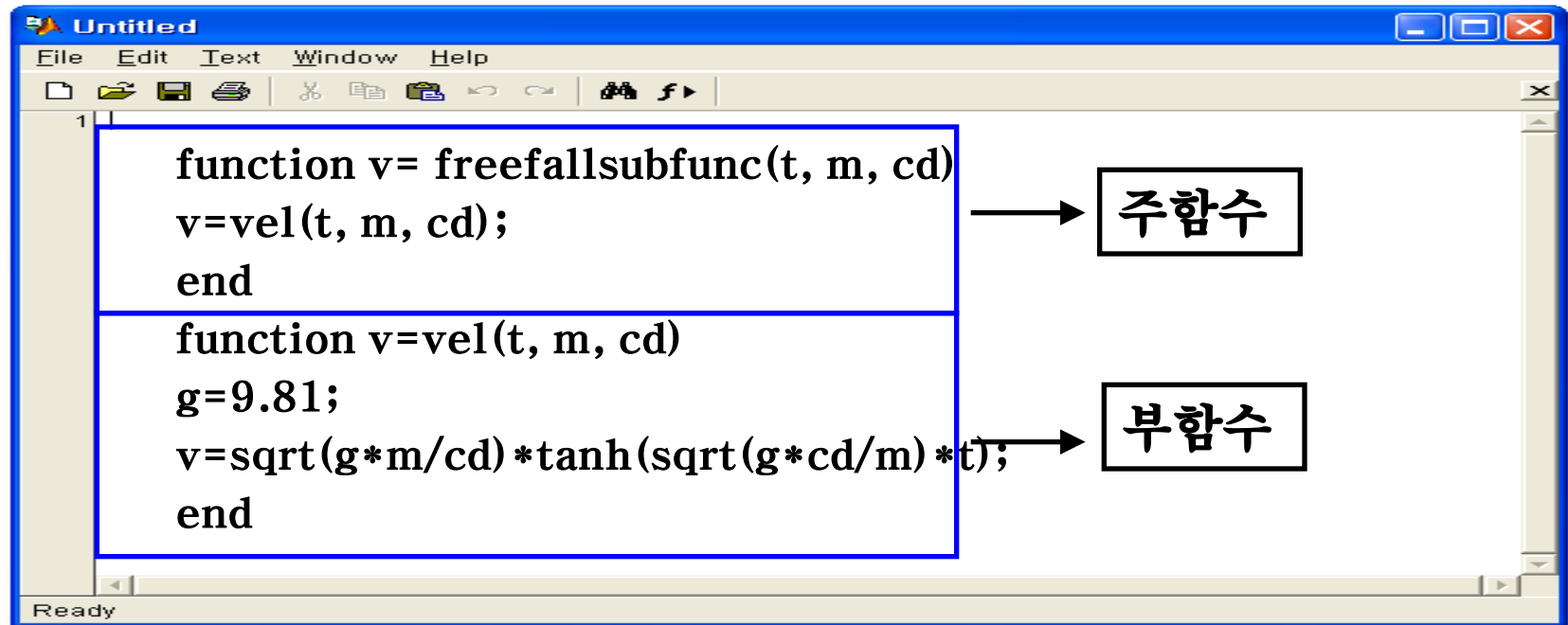




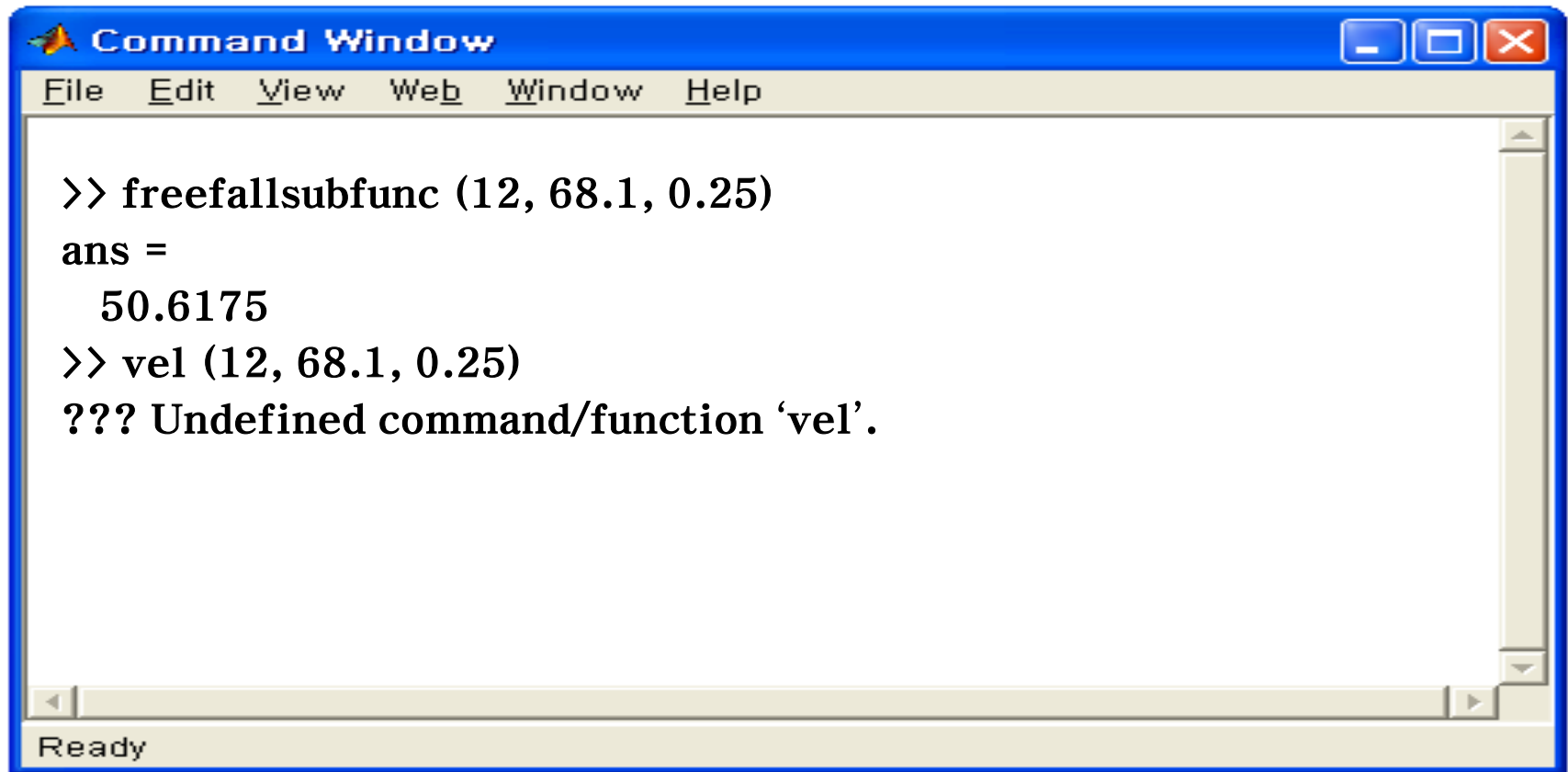
## 3.1 M-파일 (4/5)

### ■ 부함수(subfunctions)

- 함수가 다른 함수를 부를 수 있다. 이러한 함수는 M-파일을 구분하여 작성할 수도 있고, 한 개의 M 파일에 포함시킬 수도 있다.



## 3.1 M-파일 (5/5)

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window contains the following text:

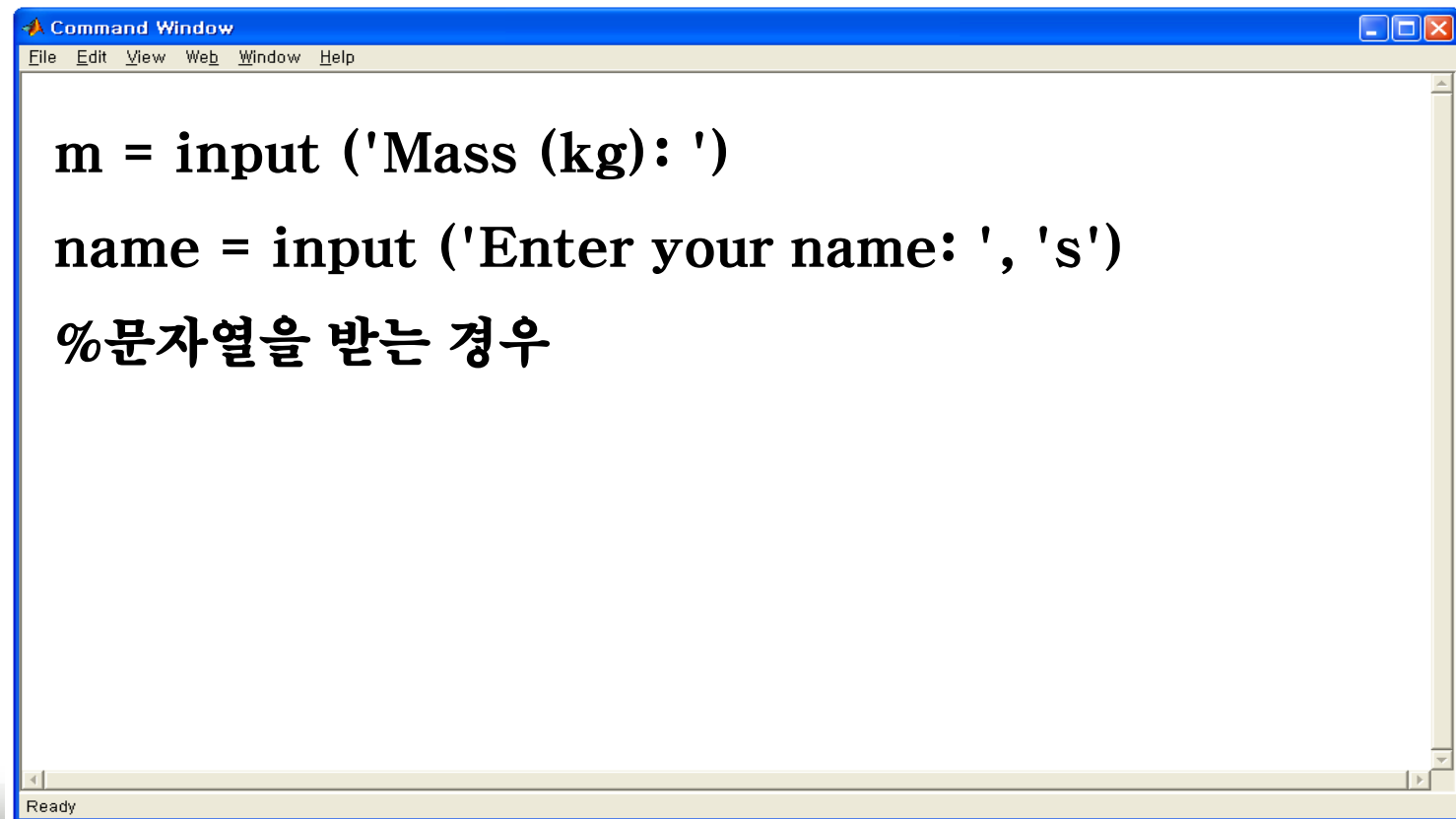
```
>> freefallsubfunc (12, 68.1, 0.25)
ans =
    50.6175
>> vel (12, 68.1, 0.25)
??? Undefined command/function 'vel'.
```

At the bottom of the window, there is a status bar that says "Ready".

## 3.2 입력-출력 (1/4)

### ■ input 함수

- 사용자로 하여금 명령창에서 직접 입력하도록 한다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text 'Command Window' and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main area of the window is white and contains the following MATLAB code:

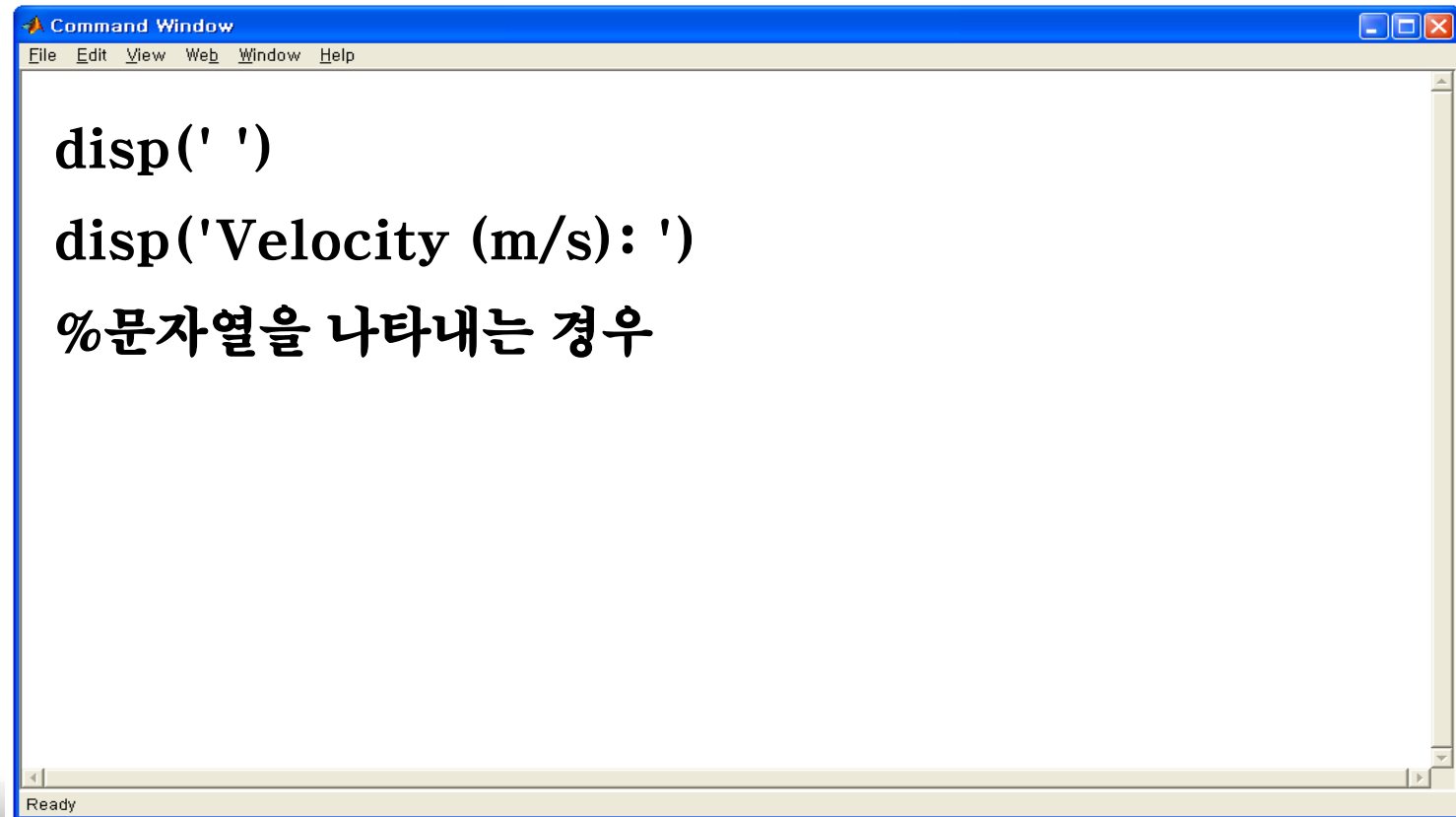
```
m = input ('Mass (kg): ')  
name = input ('Enter your name: ', 's')  
%문자열을 받는 경우
```

The status bar at the bottom of the window shows 'Ready'.

## 3.2 입력-출력 (2/4)

### ■ disp 함수

- 어떤 값을 손쉽게 나타낸다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, View, Web, Window, and Help. The main area of the window is white and contains three lines of text:

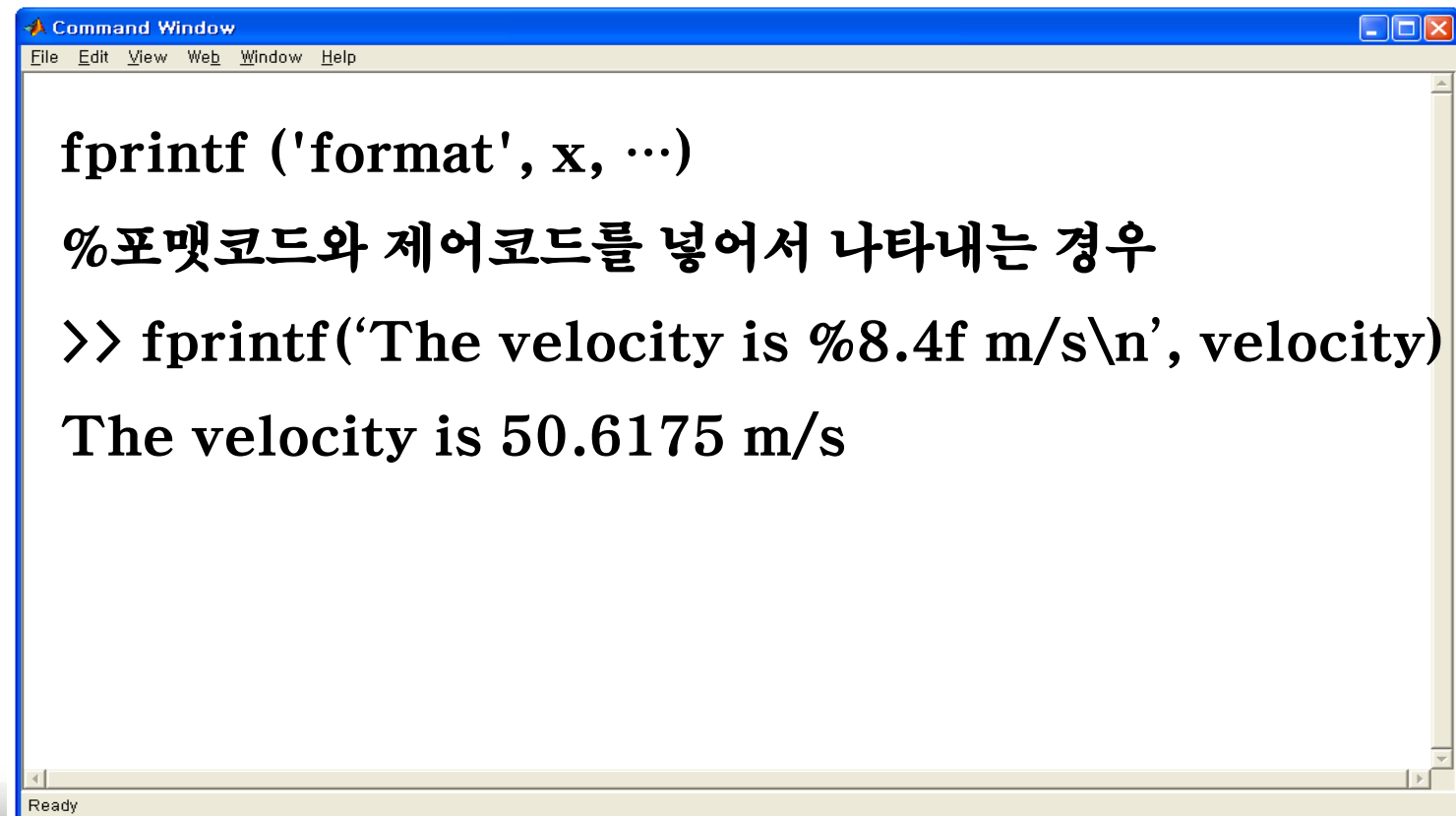
```
disp(' ')  
disp('Velocity (m/s): ')  
%문자열을 나타내는 경우
```

At the bottom of the window, there is a status bar with the text "Ready".

## 3.2 입력-출력 (3/4)

### ■ fprintf 함수

- 정보를 표현할 때 추가적인 제어를 제공한다.

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The menu bar includes "File", "Edit", "View", "Web", "Window", and "Help". The command prompt shows the function call `fprintf('format', x, ...)` followed by a description in Korean: "%포맷코드와 제어코드를 넣어서 나타내는 경우". Then, the command `>> fprintf('The velocity is %8.4f m/s\n', velocity)` is entered, and the output "The velocity is 50.6175 m/s" is displayed. The status bar at the bottom left says "Ready".

```
Command Window
File Edit View Web Window Help

fprintf('format', x, ...)
%포맷코드와 제어코드를 넣어서 나타내는 경우
>> fprintf('The velocity is %8.4f m/s\n', velocity)
The velocity is 50.6175 m/s

Ready
```



## 3.2 입력-출력 (4/4)

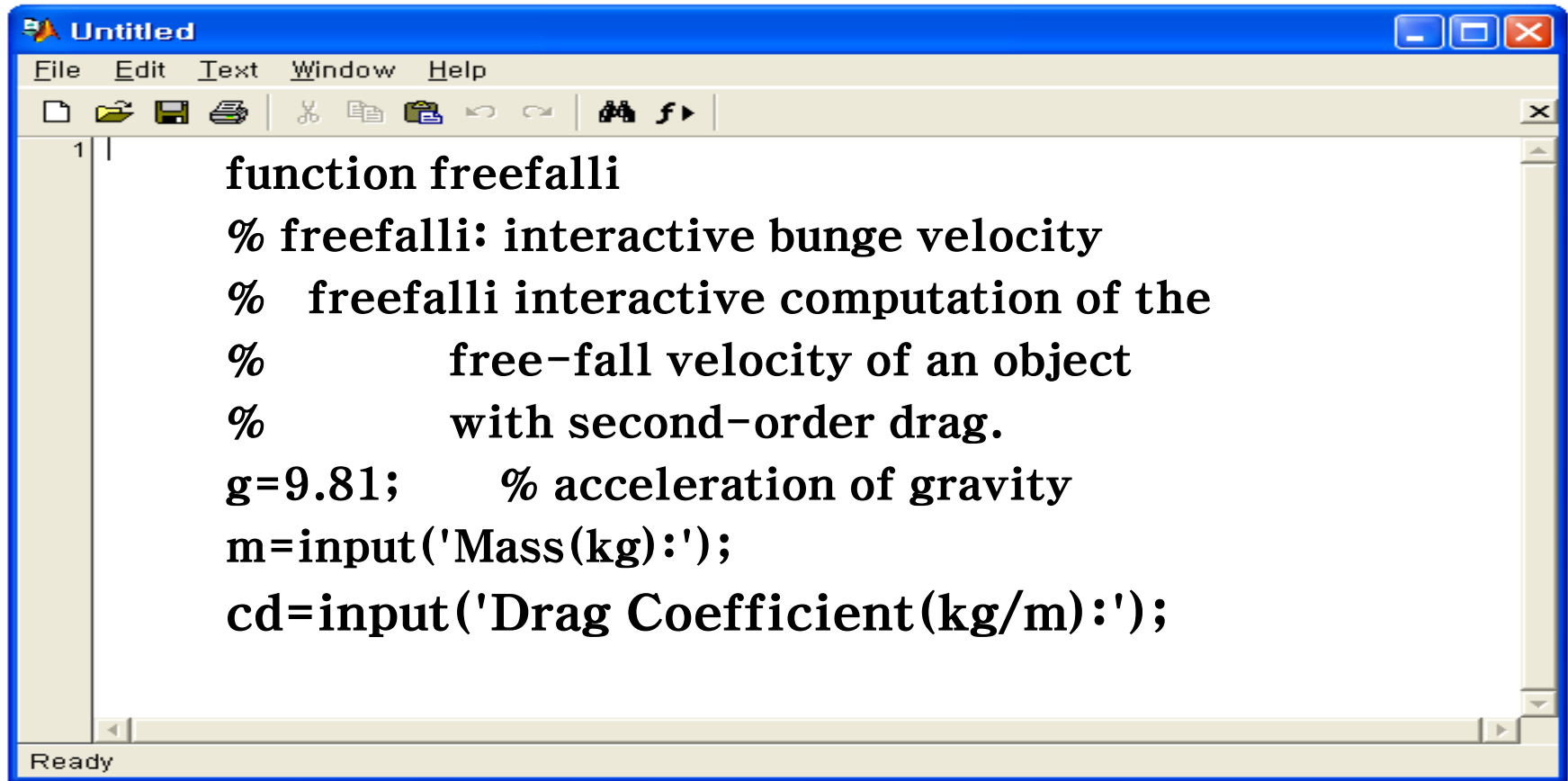
표 3.1 fprintf 함수에서 사용하는 포맷 코드와 제어 코드

포맷 코드	설 명
%d	정수 포맷
%e	e를 사용하는 과학 포맷
%E	E를 사용하는 과학 포맷
%f	소수 포맷
%g	%e나 %f 중 간단한 포맷
제어 코드	설 명
\n	새로운 줄로 시작
\t	탭



## 예제 3.3 [1/3]

Q. 예제 3.2에서와 같이 번지점프하는 사람의 자유낙하 속도를 계산하라.  
입출력으로 input과 disp 함수를 사용하라.



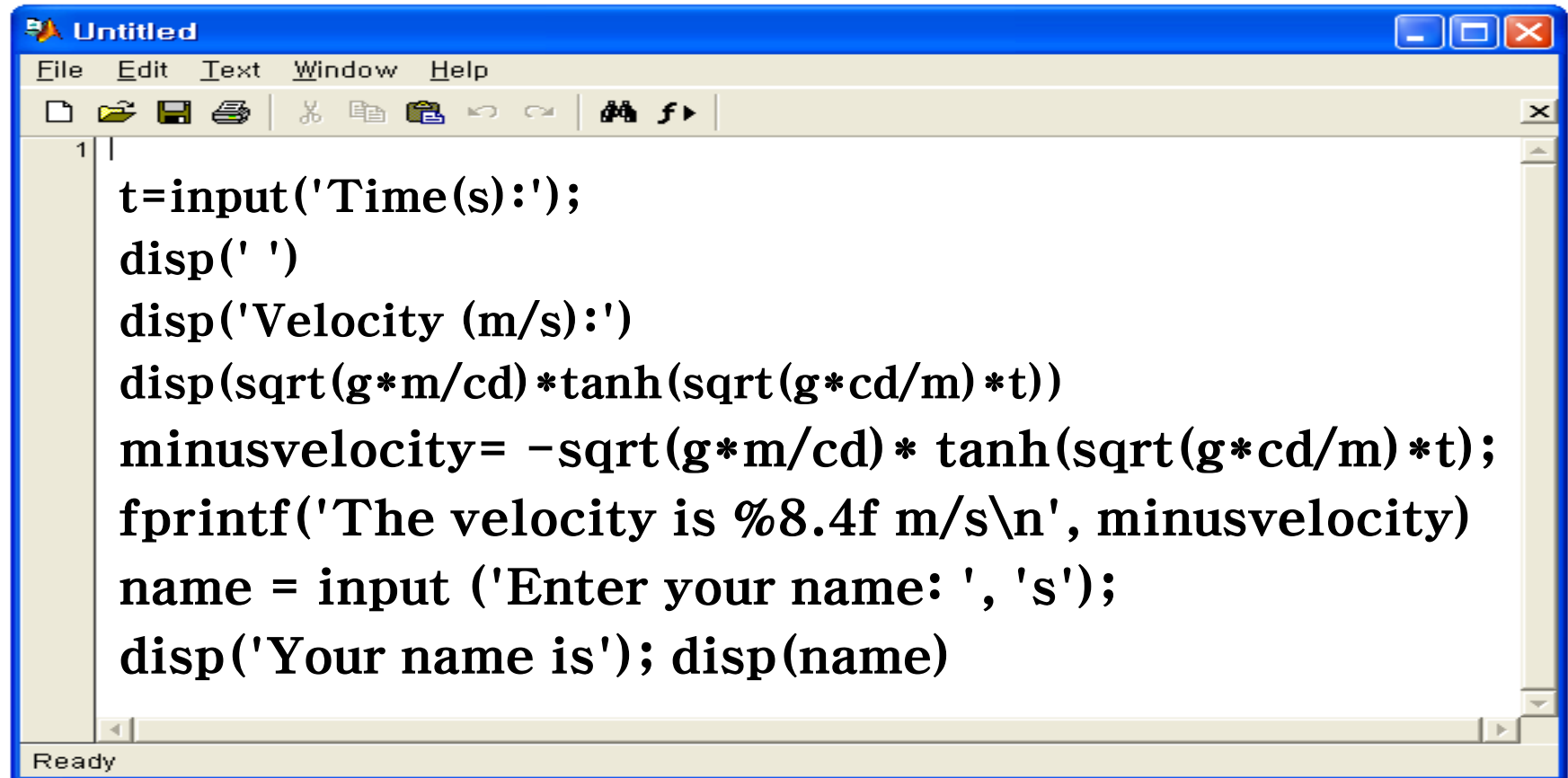
```
function freefalli
% freefalli: interactive bunge velocity
% freefalli interactive computation of the
% free-fall velocity of an object
% with second-order drag.
g=9.81; % acceleration of gravity
m=input('Mass(kg):');
cd=input('Drag Coefficient(kg/m):');
```





## 예제 3.3 [2/3]

Q. 예제 3.2에서와 같이 번지점프하는 사람의 자유낙하 속도를 계산하라.  
입출력으로 input과 disp 함수를 사용하라.



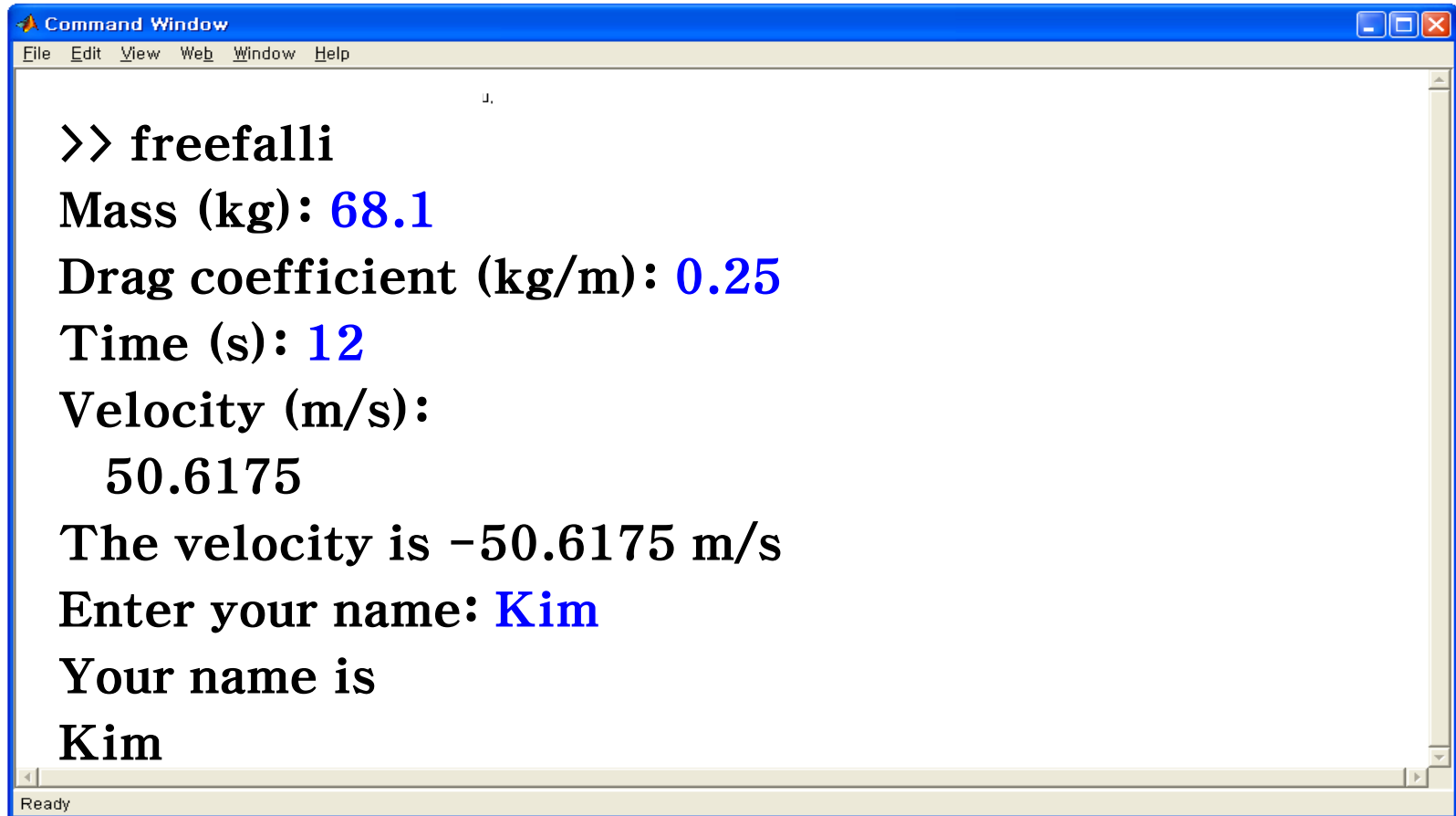
```
Untitled
File Edit Text Window Help
[Icons]
1 |
t=input('Time(s):');
disp(' ')
disp('Velocity (m/s):')
disp(sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t))
minusvelocity= -sqrt(g*m/cd)* tanh(sqrt(g*cd/m)*t);
fprintf('The velocity is %8.4f m/s\n', minusvelocity)
name = input ('Enter your name: ', 's');
disp('Your name is'); disp(name)
```

Ready



## 예제 3.3 [3/3]

- 명령창에서 다음과 같이 입력한다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text 'Command Window' and standard window controls. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main area of the window contains the following text: '>> freefalli', 'Mass (kg): 68.1', 'Drag coefficient (kg/m): 0.25', 'Time (s): 12', 'Velocity (m/s):', '50.6175', 'The velocity is -50.6175 m/s', 'Enter your name: Kim', 'Your name is', 'Kim'. The status bar at the bottom left says 'Ready'.

```
>> freefalli
Mass (kg): 68.1
Drag coefficient (kg/m): 0.25
Time (s): 12
Velocity (m/s):
50.6175
The velocity is -50.6175 m/s
Enter your name: Kim
Your name is
Kim
```



## 3.3 구조 프로그래밍 (1/12)

### ■ 명령을 연속적으로 수행하지 않는 것을 허용하는 구문

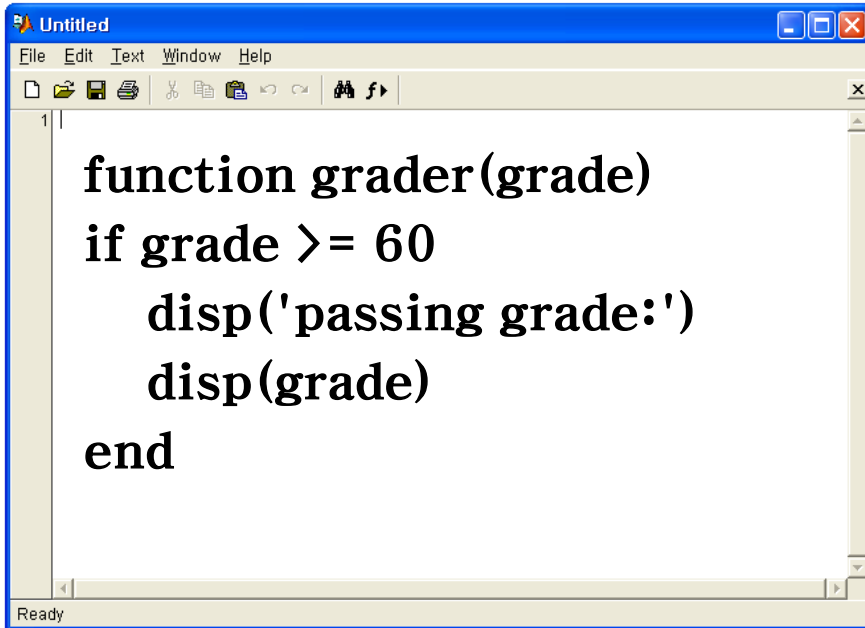
- 판정 (또는 선택): 판정에 기초를 둔 흐름의 분기점이다.
- 루프 (또는 반복): 반복을 허용하는 흐름의 루프이다.

### ■ 판정 [if 구조]

```
if condition  
  statements  
end
```



## 3.3 구조 프로그래밍 (2/12)

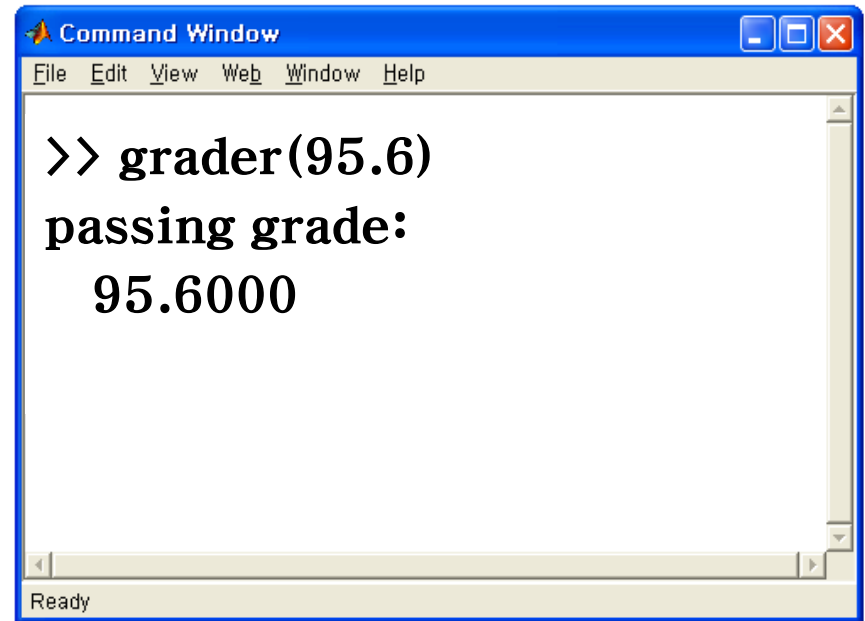


The image shows a MATLAB 'Untitled' editor window. The title bar is blue with the MATLAB logo and the text 'Untitled'. The menu bar includes 'File', 'Edit', 'Text', 'Window', and 'Help'. The toolbar contains icons for file operations (new, open, save, print) and editing (undo, redo, cut, copy, paste, find). The main text area contains the following MATLAB code:

```
1 function grader(grade)
   if grade >= 60
       disp('passing grade:')
       disp(grade)
   end
```

The status bar at the bottom left says 'Ready'.

<파일 편집기>



The image shows a MATLAB 'Command Window'. The title bar is blue with the MATLAB logo and the text 'Command Window'. The menu bar includes 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main text area shows the execution of the 'grader' function with the input 95.6:

```
>> grader(95.6)
passing grade:
    95.6000
```

The status bar at the bottom left says 'Ready'.

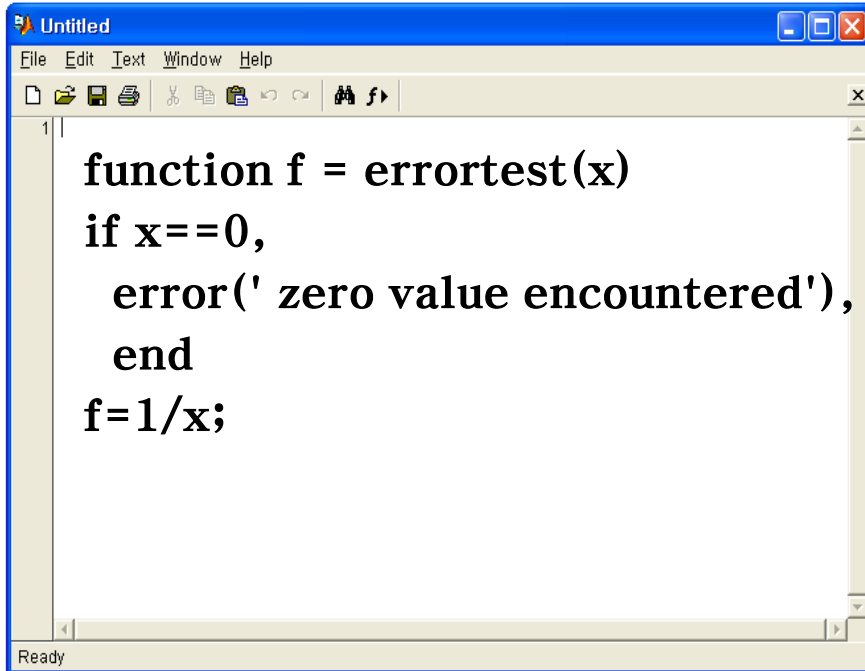
<명령창>



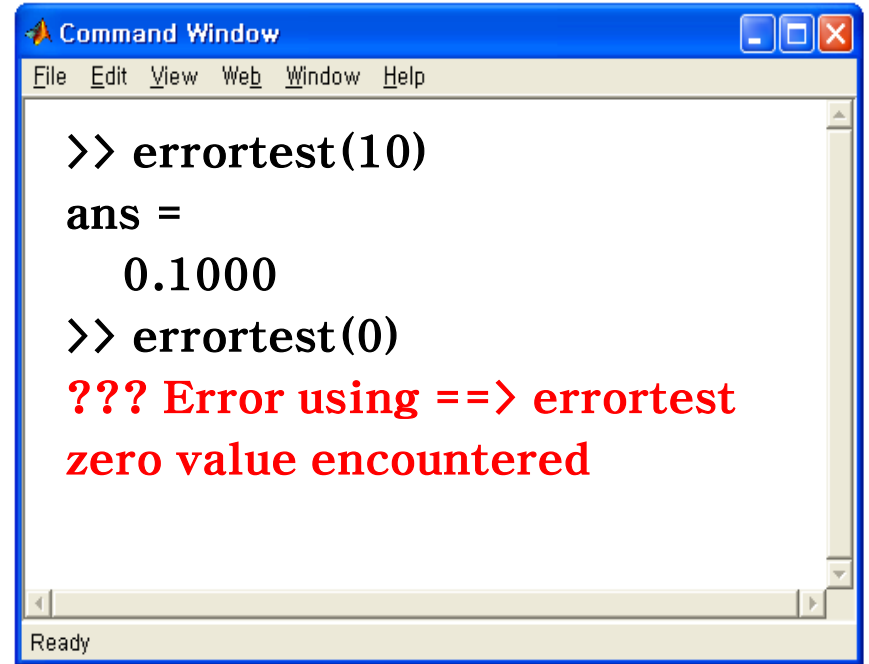
## 3.3 구조 프로그래밍 (3/12)

### [에러 함수]

`error ( msg )`



```
function f = errortest(x)
if x==0,
    error(' zero value encountered'),
end
f=1/x;
```



```
>> errortest(10)
ans =
    0.1000
>> errortest(0)
??? Error using ==> errortest
zero value encountered
```



## 3.3 구조 프로그래밍 (4/12)

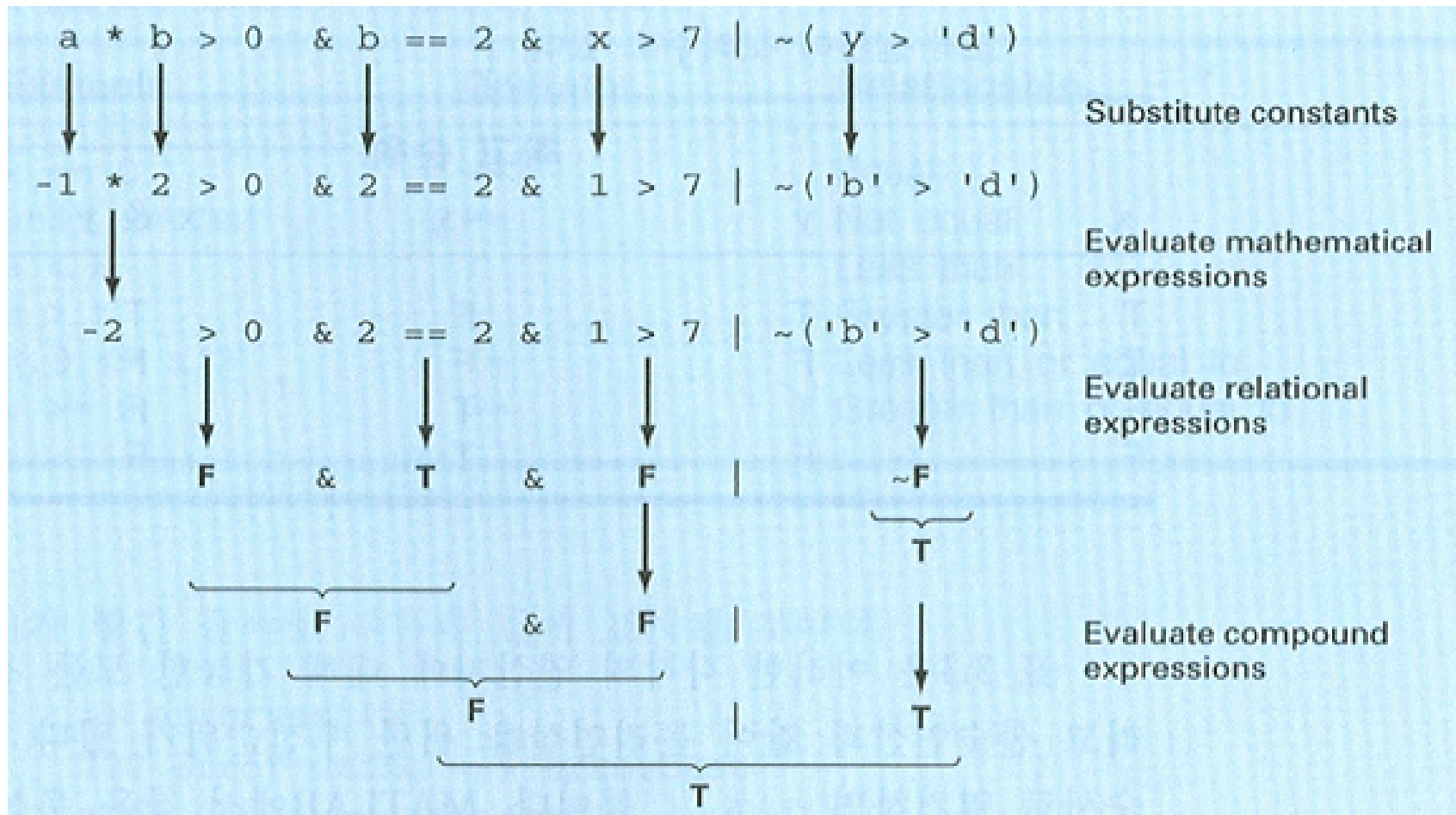
### [논리조건]

`value1` *relation* `value2`

- ~ (Not)
  - 논리적 부정을 나타낼 때 사용한다.
- & (And)
  - 두 식에서 논리적 곱을 나타낼 때 사용한다.
- | (Or)
  - 두 식에서 논리적 합을 나타낼 때 사용한다.



## 3.3 구조 프로그래밍 (5/12)



<복잡한 논리식의 단계별 계산>





## 3.3 구조 프로그래밍 (6/12)

### [if ... else 구조]

```
if condition
    statements1
else
    statements2
end
```

### [if ... elseif 구조]

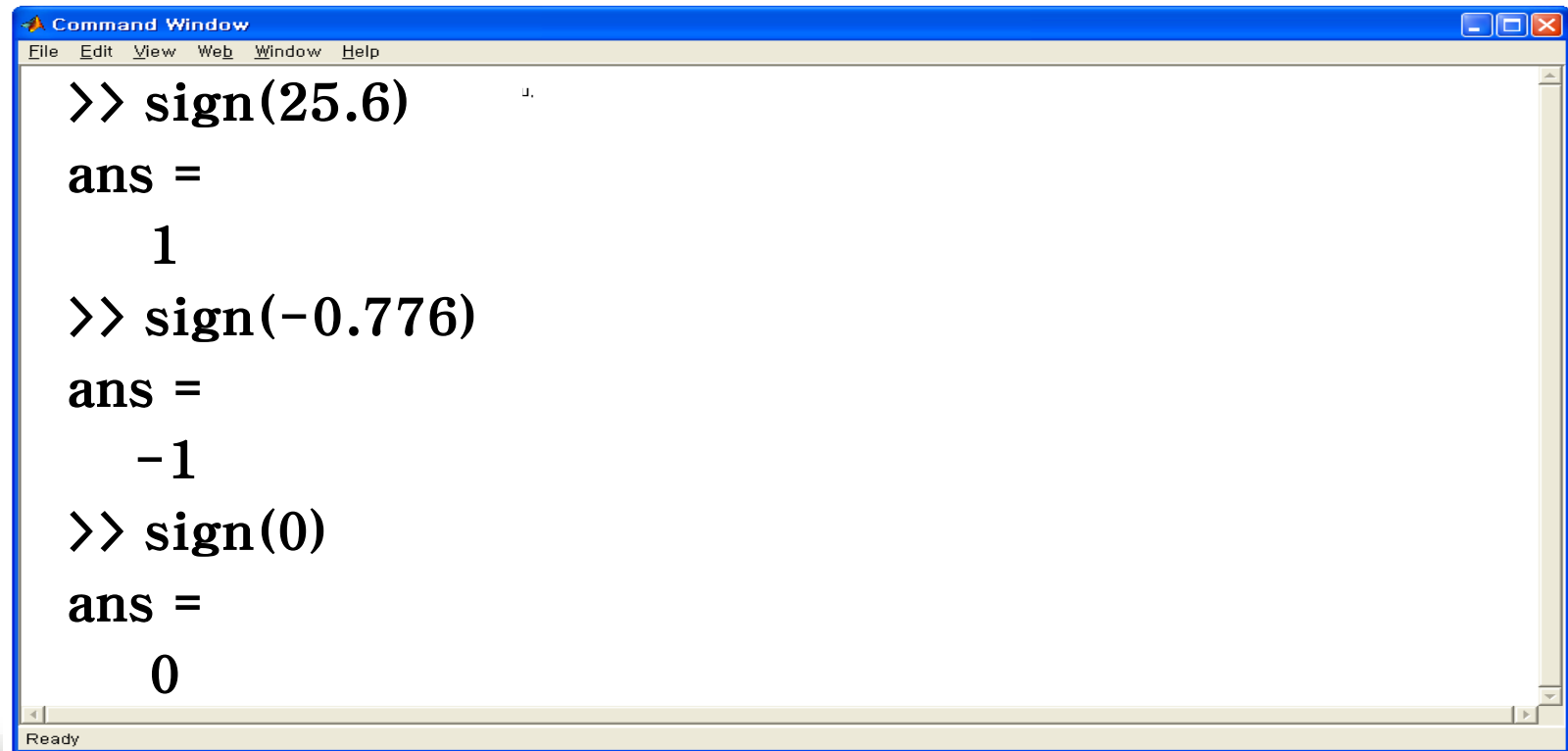
```
if condition1
    statements1
elseif condition2
    statements2
...
else
    statementselse
end
```



## 예제 3.4 [1/3]

Q. 내장함수인 sign 함수와 같은 기능을 갖도록  
if ...else 구조를 사용하여 mysign 함수를 작성하라.

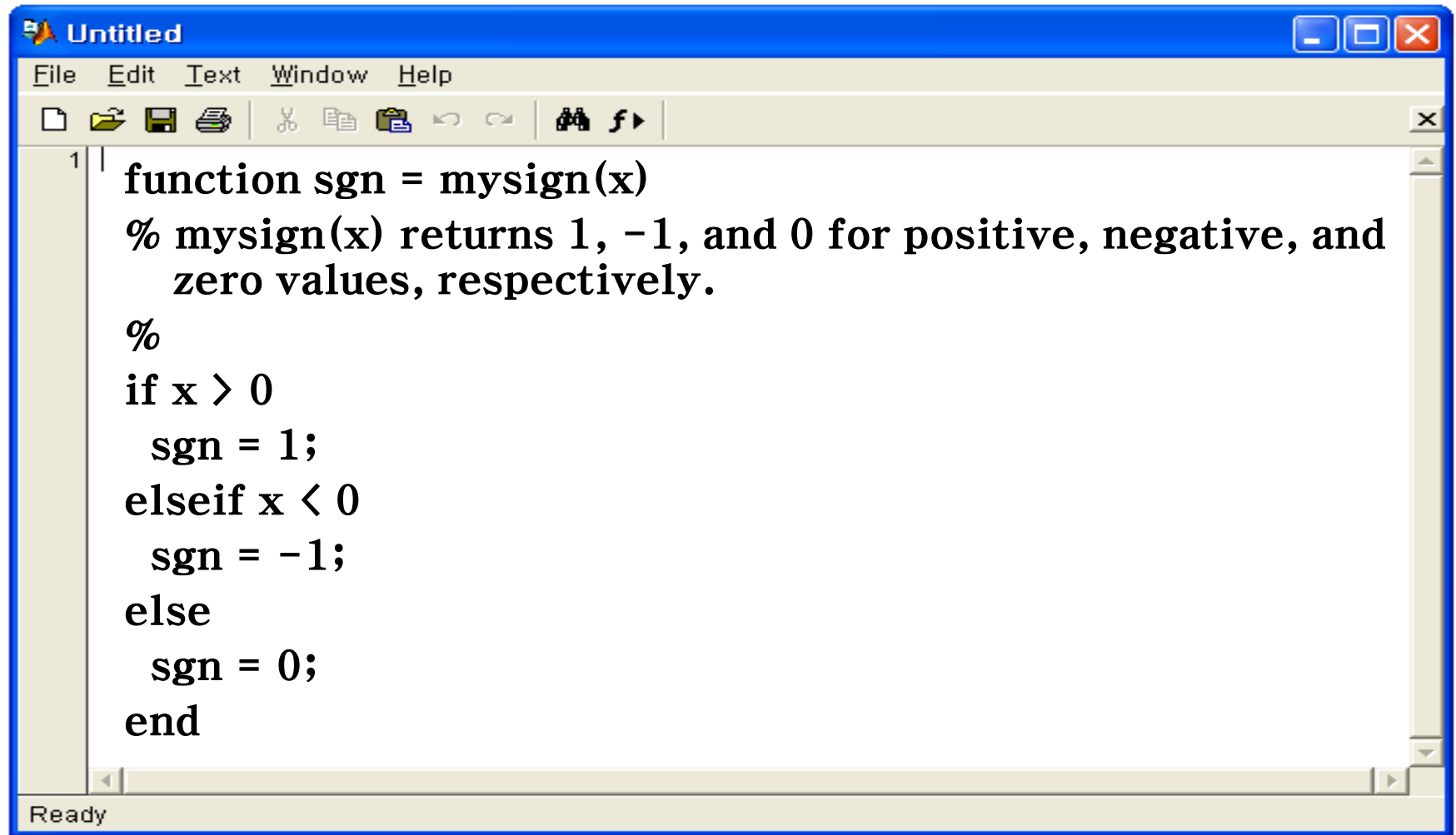
풀이) 내장함수인 sign 함수의 기능을 알아보자.



```
Command Window
File Edit View Web Window Help
>> sign(25.6)
ans =
    1
>> sign(-0.776)
ans =
   -1
>> sign(0)
ans =
    0
Ready
```



## 예제 3.4 [2/3]

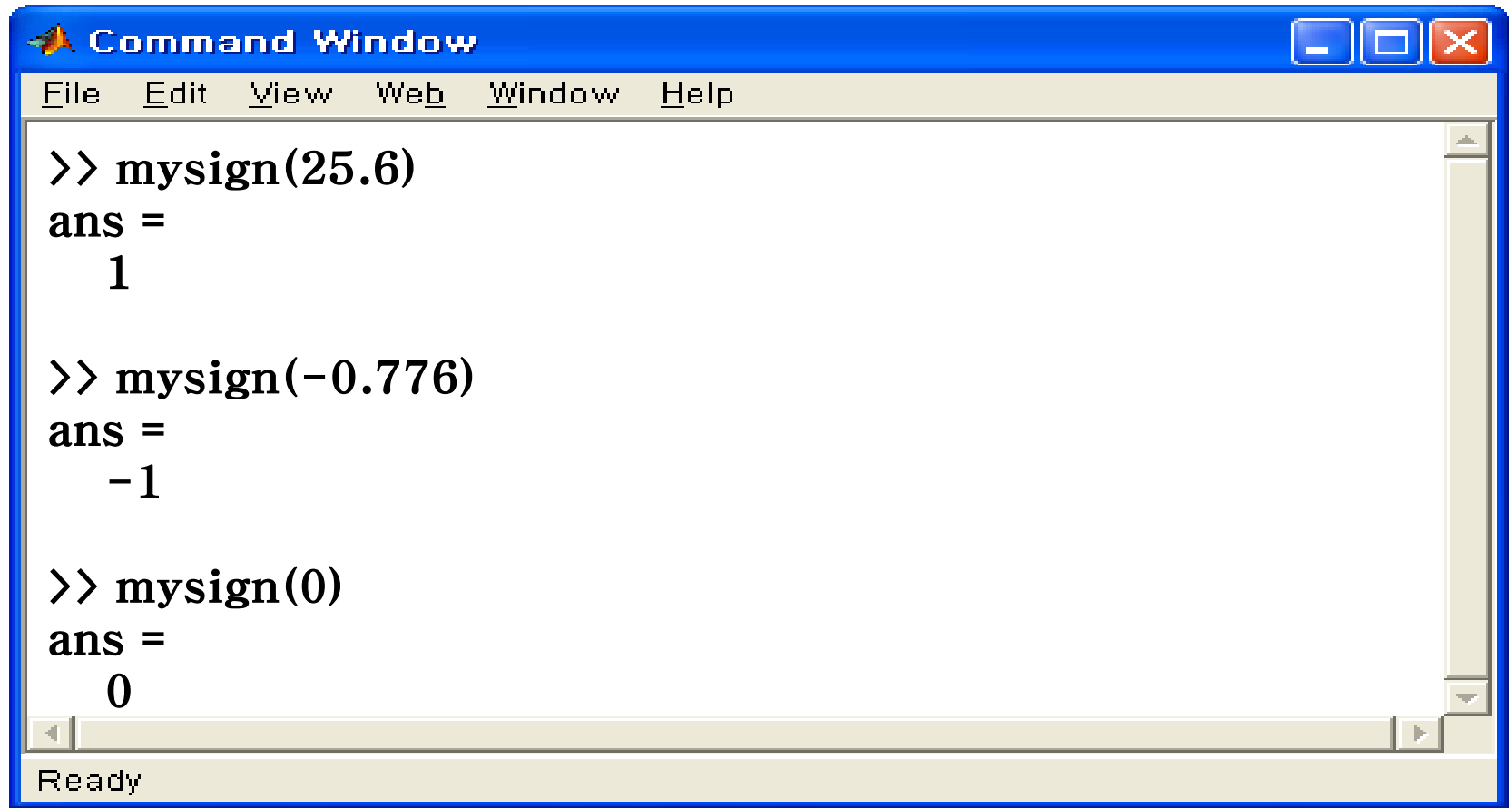
A screenshot of a MATLAB 'Untitled' window. The window has a blue title bar with standard Windows controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Text', 'Window', and 'Help'. Underneath the menu bar is a toolbar with icons for file operations (new, open, save, print), editing (cut, copy, paste), and MATLAB-specific functions (run, save all). The main text area contains a MATLAB script for a sign function. The script starts with a function definition line, followed by a comment explaining the function's behavior. It then uses an 'if-elseif-else' structure to assign the sign of the input 'x' to the variable 'sgn'. The status bar at the bottom of the window displays the word 'Ready'.

```
1 function sgn = mysign(x)
% mysign(x) returns 1, -1, and 0 for positive, negative, and
% zero values, respectively.
%
if x > 0
    sgn = 1;
elseif x < 0
    sgn = -1;
else
    sgn = 0;
end
```



## 예제 3.4 [3/3]

명령창에서 다음과 같이 확인할 수 있다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, View, Web, Window, and Help. The main area of the window is white and contains three lines of MATLAB code and their corresponding outputs. The first line is ">> mysign(25.6)" followed by "ans = 1". The second line is ">> mysign(-0.776)" followed by "ans = -1". The third line is ">> mysign(0)" followed by "ans = 0". At the bottom of the window is a status bar with the text "Ready".

```
Command Window
File Edit View Web Window Help

>> mysign(25.6)
ans =
    1

>> mysign(-0.776)
ans =
   -1

>> mysign(0)
ans =
    0

Ready
```



## 3.3 구조 프로그래밍 (7/12)

### ■ 루프

[for ... end 구조]

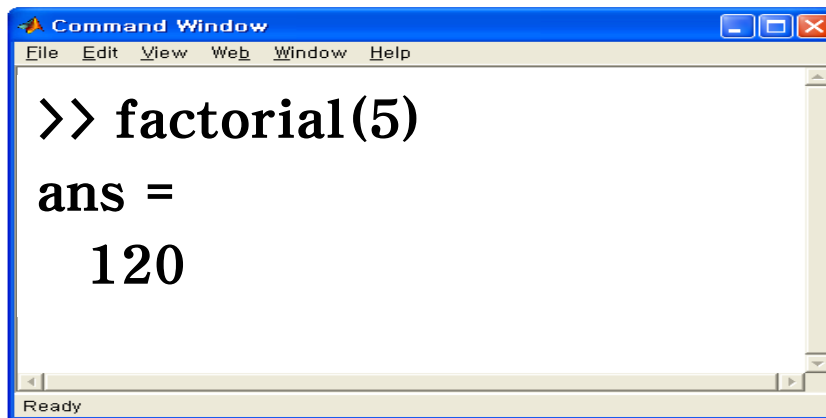
```
for index = start:step:finish  
    statements  
end
```



## 예제 3.5 [1/3]

Q. 순차곱셈을 위한 for 루프

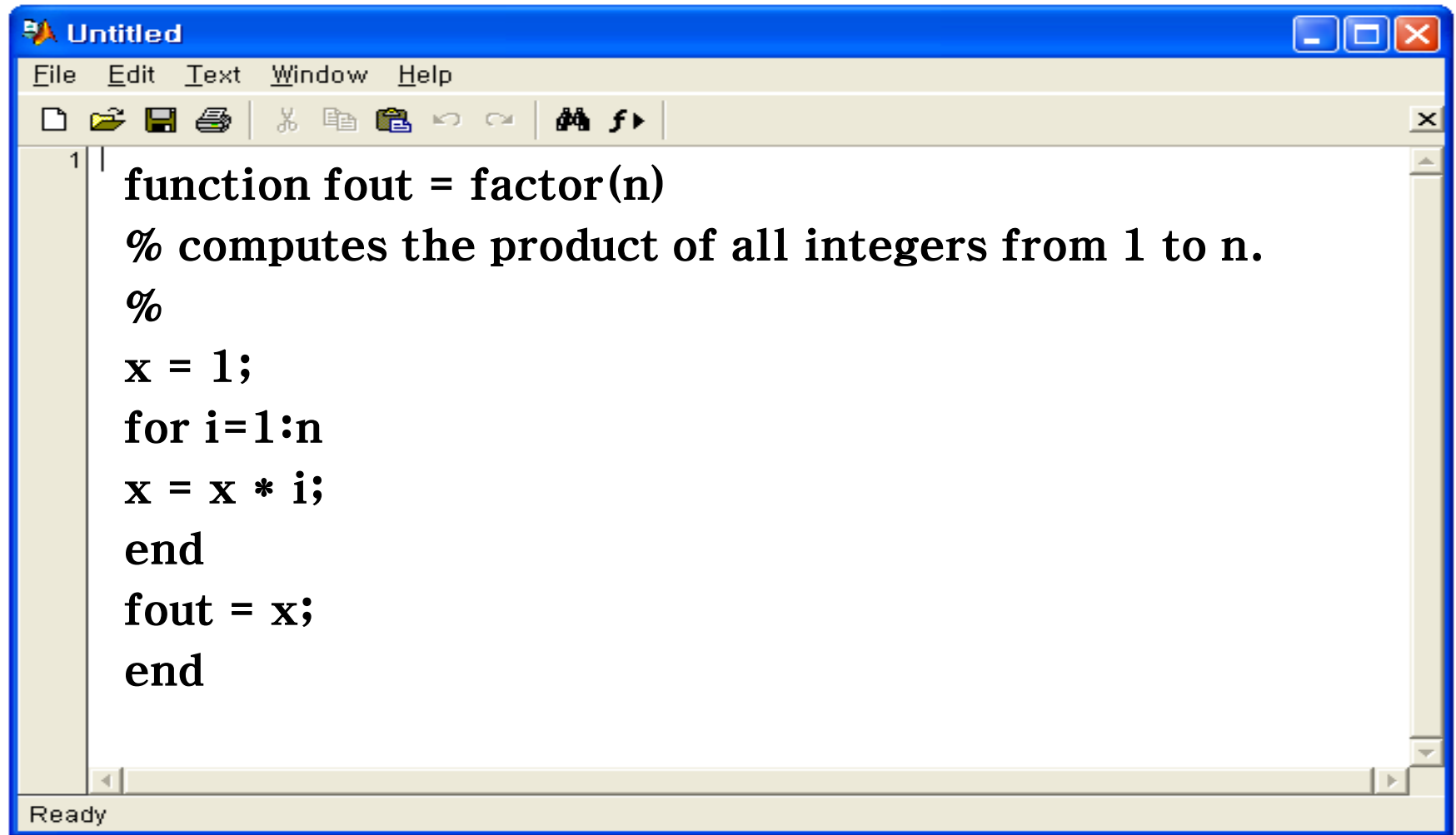
풀이) factorial 함수와 같은 기능을 갖도록 for 루프를 사용하여 프로그램을 작성한다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window is white and contains the text: ">> factorial(5)" followed by "ans =" and "120". At the bottom of the window, there is a status bar that says "Ready".

```
>> factorial(5)
ans =
    120
```



## 예제 3.5 [2/3]



A screenshot of a MATLAB 'Untitled' window. The window has a blue title bar with standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Text', 'Window', and 'Help'. Underneath the menu bar is a toolbar with icons for file operations (new, open, save, print), editing (cut, copy, paste), and execution (run, stop). The main text area contains a MATLAB script for a factorial function. The script starts with a line number '1' in the left margin. The code is as follows:

```
function fout = factor(n)
% computes the product of all integers from 1 to n.
%
x = 1;
for i=1:n
x = x * i;
end
fout = x;
end
```

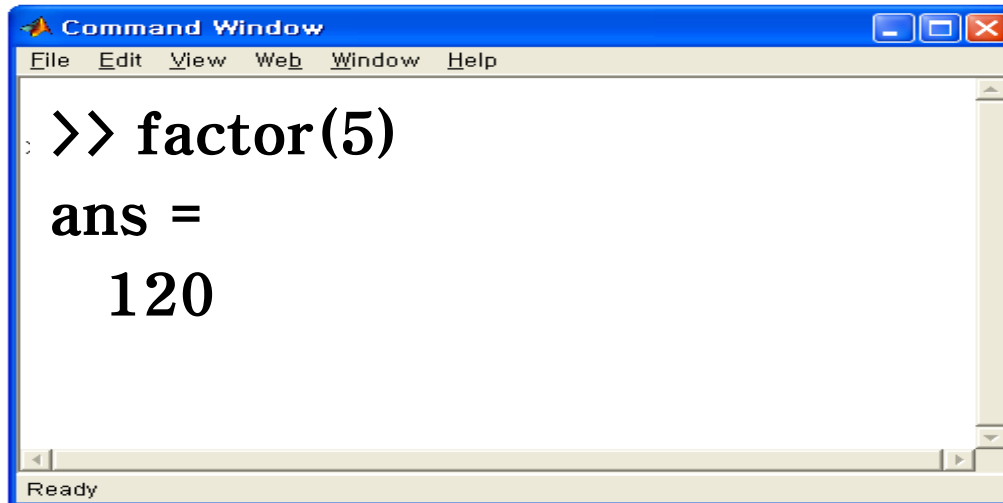
At the bottom of the window is a status bar that says 'Ready'.





## 예제 3.5 (3/3)

명령창에서 다음과 같이 확인할 수 있다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with the following options: File, Edit, View, Web, Window, and Help. The main area of the window is white and contains the text: ">> factor(5)" on the first line, "ans =" on the second line, and "120" on the third line. At the bottom of the window, there is a status bar with the text "Ready".

```
>> factor(5)
ans =
    120
```



## 3.3 구조 프로그래밍 (8/12)

### [벡터화]

```
i = 0;  
for t = 0:0.02:5  
    i = i + 1;  
    y(i) = cos(t);  
end
```

위와 같은 for 루프를 다음과 같이 벡터화할 수 있다.

```
t = 0:0.02:5;  
y = cos(t);
```



## 3.3 구조 프로그래밍 (9/12)

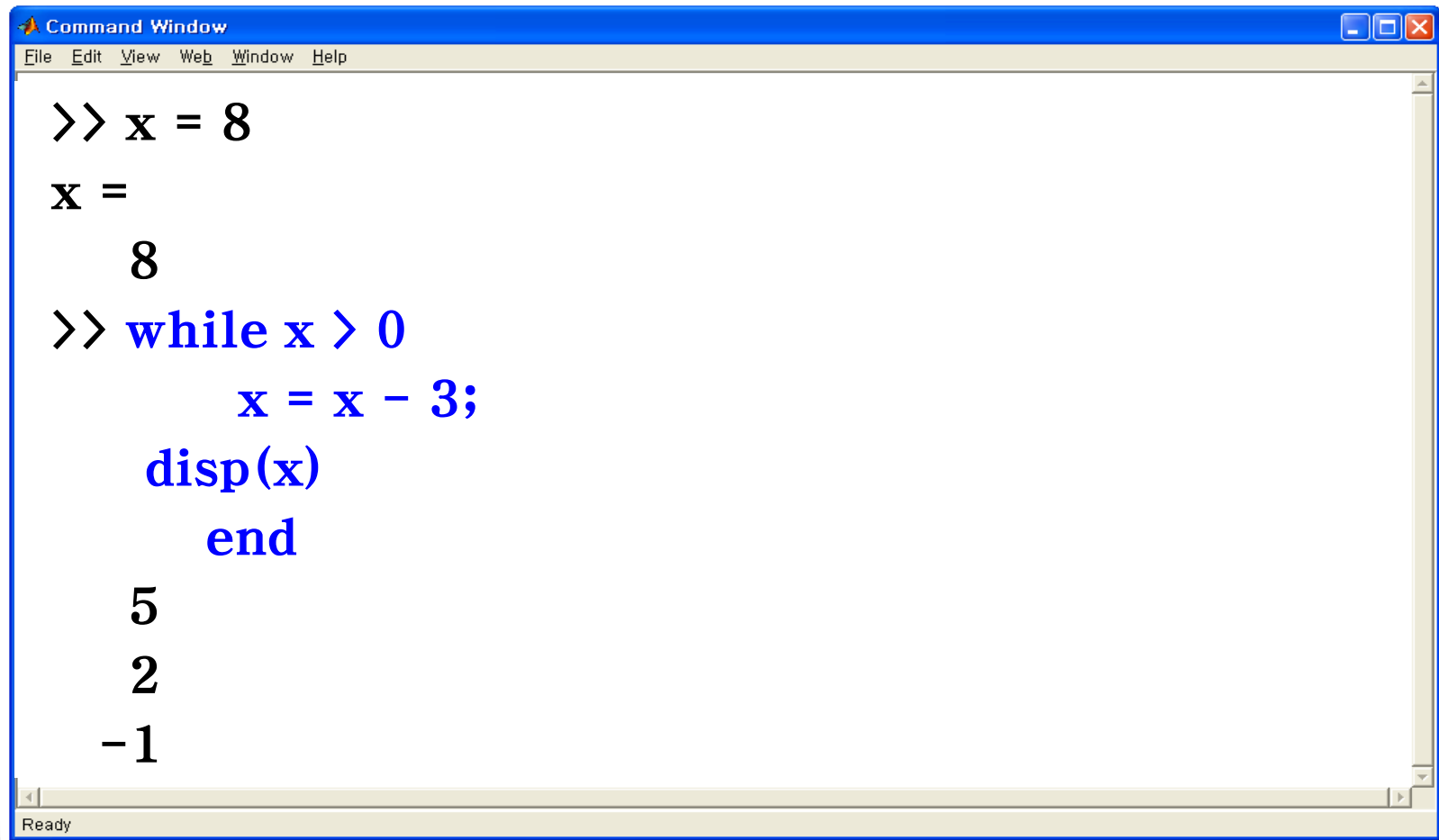
### [while 구조]

```
while condition  
    statements  
end
```

while과 end 사이의 조건이 참인 동안에만 반복한다.  
(while 루프의 종료는 시작 위치에서 조건이 맞지 않을 경우 발생한다.)



## 3.3 구조 프로그래밍 (10/12)

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text 'Command Window' and standard window controls. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main area of the window contains MATLAB code and its output. The code starts with 'x = 8', followed by a 'while' loop that decrements 'x' by 3 and displays its value until it is no greater than 0. The output shows the values 8, 5, 2, and -1. The status bar at the bottom left says 'Ready'.

```
>> x = 8
x =
     8
>> while x > 0
        x = x - 3;
        disp(x)
    end
     5
     2
    -1
```



## 3.3 구조 프로그래밍 (11/12)

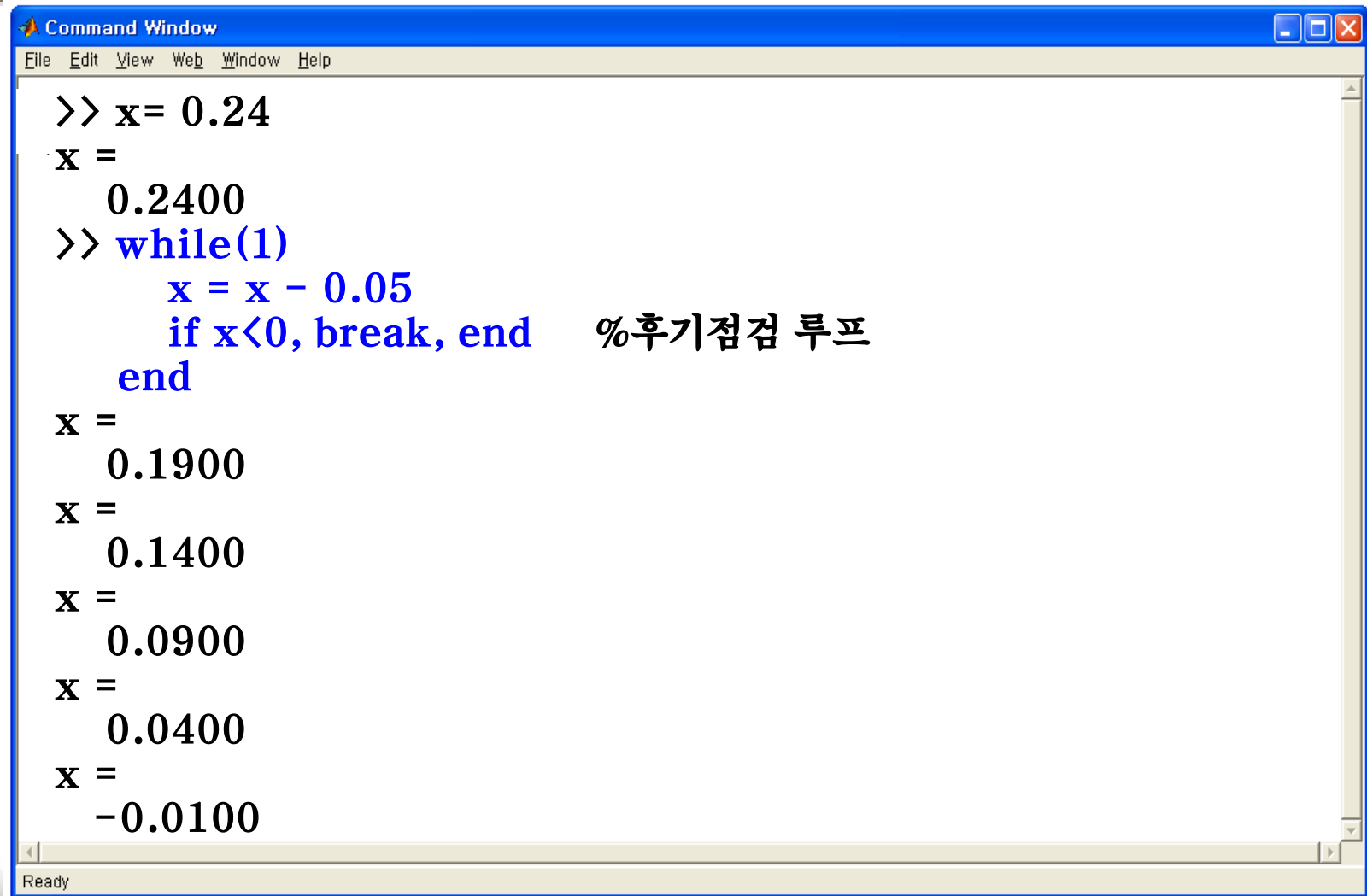
### [while ... break 구조]

```
while (1)
    statements
    if condition, break, end
        statements
    end
```

while과 end 사이의 조건이 참인 동안 반복하나, while 루프 내의 어떤 위치에서도 종료할 수 있다. 그 예는 다음과 같다.



## 3.3 구조 프로그래밍 (12/12)



```
Command Window
File Edit View Web Window Help

>> x= 0.24
x =
    0.2400
>> while(1)
    x = x - 0.05
    if x<0, break, end    %후기점검 루프
end
x =
    0.1900
x =
    0.1400
x =
    0.0900
x =
    0.0400
x =
   -0.0100

Ready
```



## 3.4 내포화와 들여쓰기

---

### ■ 내포화

- 다른 구조 안에 구조를 배치하는 것이다.





## 예제 3.7(3.6) [1/3]

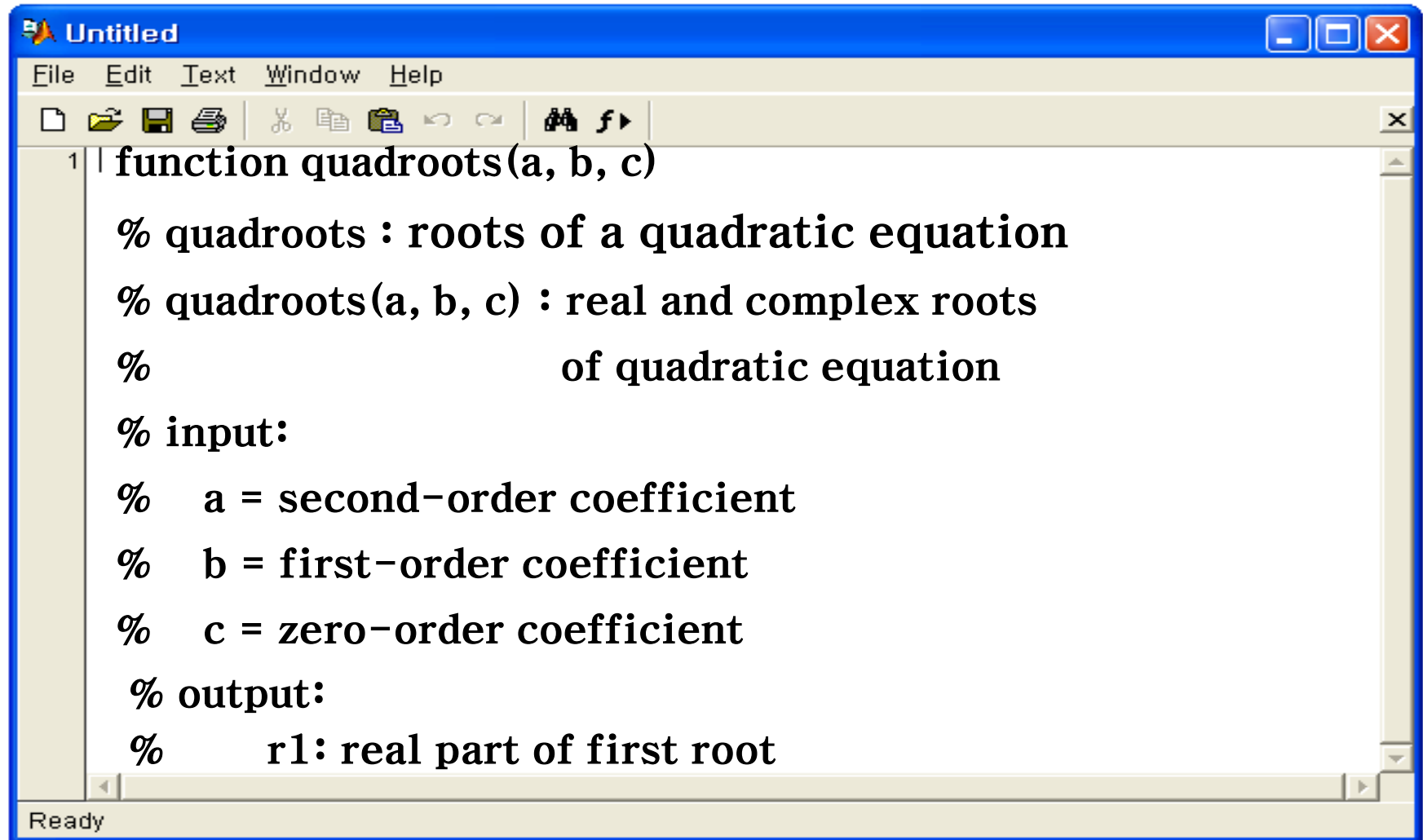
Q. 2차방정식을 풀기 위해 내포화를 이용함

$$f(x) = ax^2 + bx + c$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



## 예제 3.7(3.6) [2/3]



The image shows a MATLAB window titled 'Untitled' with a standard menu bar (File, Edit, Text, Window, Help) and a toolbar. The main text area contains the following MATLAB code:

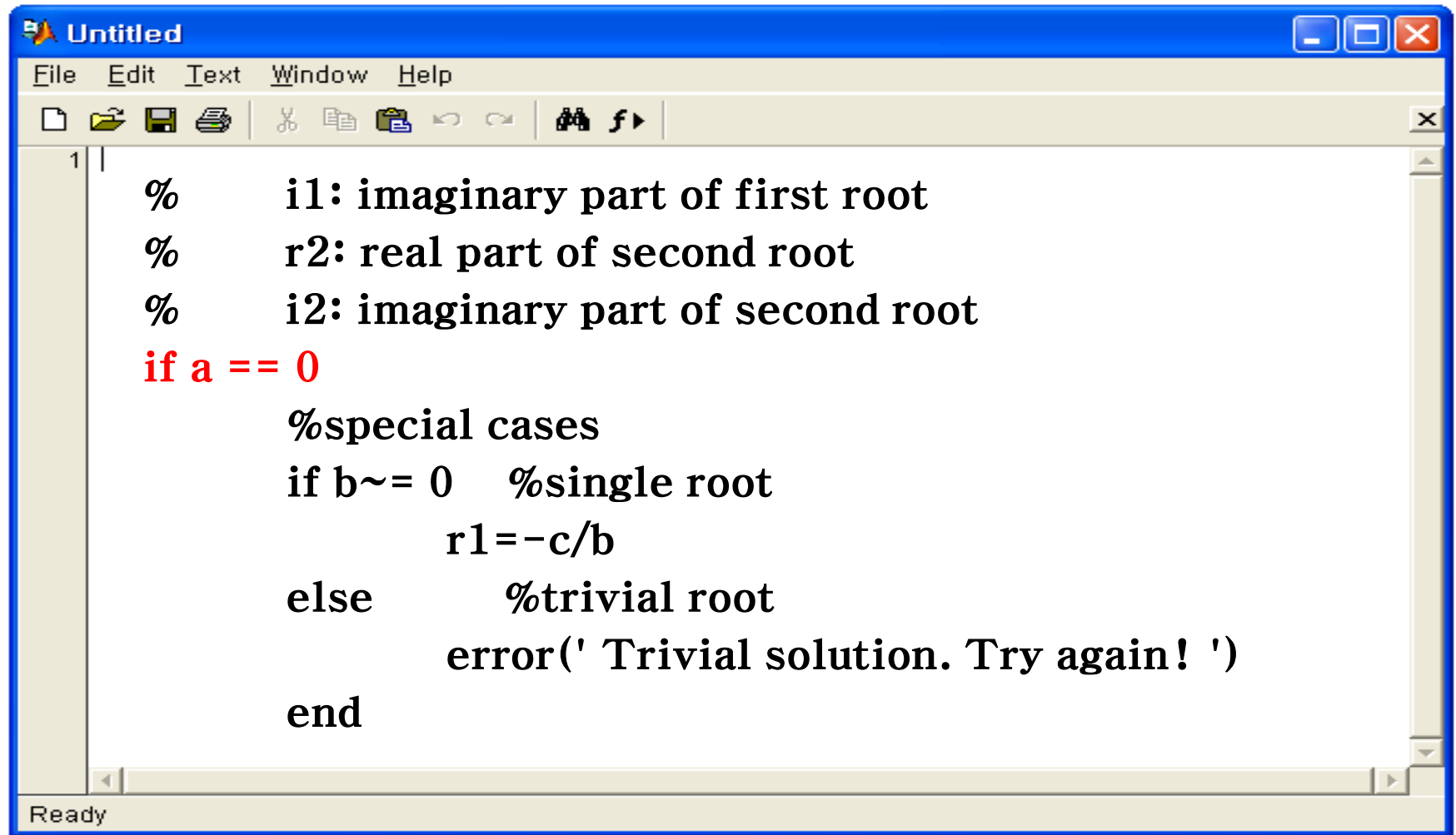
```
1 function quadroots(a, b, c)

% quadroots : roots of a quadratic equation
% quadroots(a, b, c) : real and complex roots
%                        of quadratic equation
% input:
%   a = second-order coefficient
%   b = first-order coefficient
%   c = zero-order coefficient
% output:
%   r1: real part of first root
```

The status bar at the bottom of the window displays the word 'Ready'.



## 예제 3.7(3.6) [2/3]

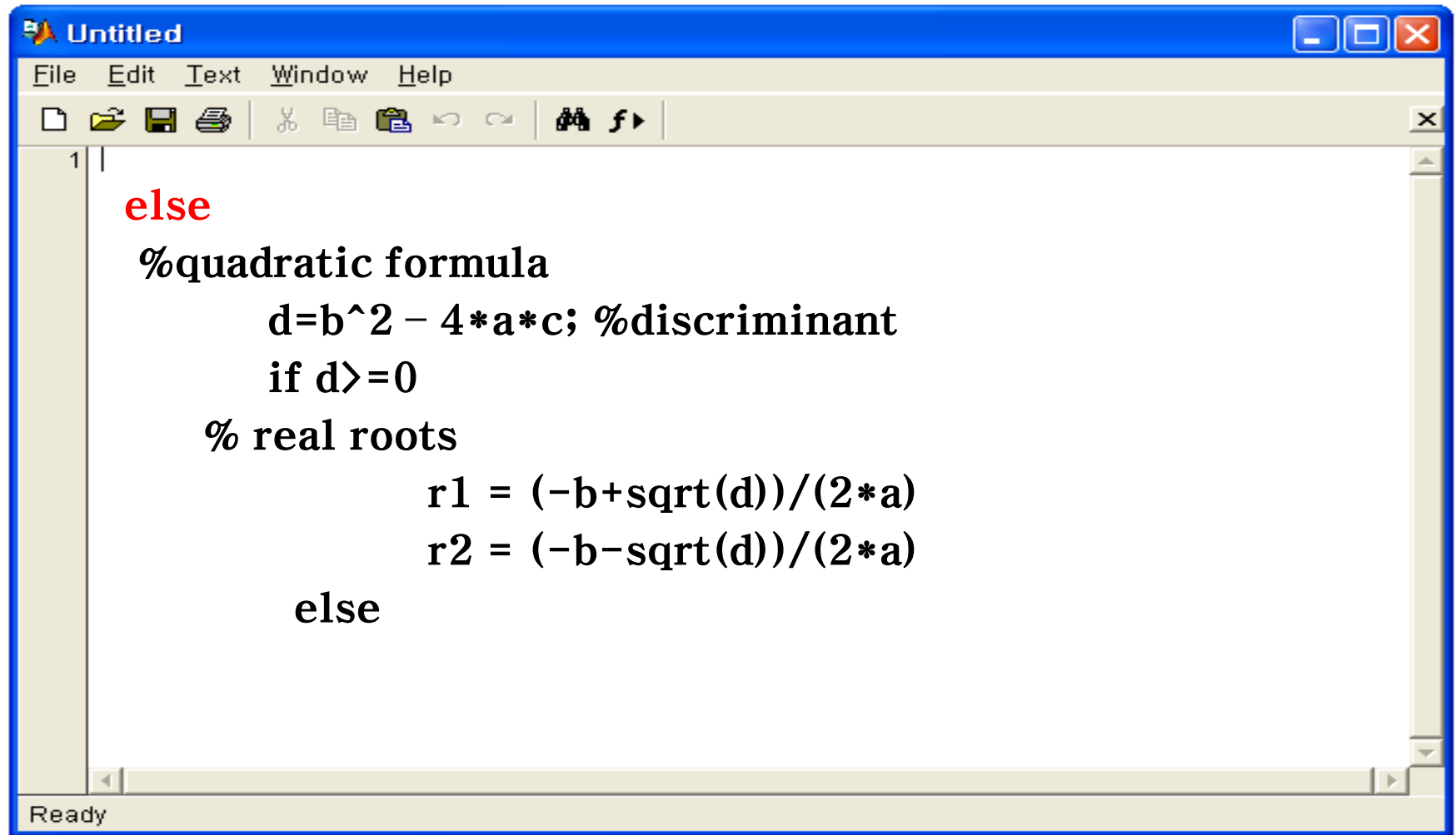


```
1 |  
% i1: imaginary part of first root  
% r2: real part of second root  
% i2: imaginary part of second root  
if a == 0  
    %special cases  
    if b~= 0 %single root  
        r1=-c/b  
    else %trivial root  
        error(' Trivial solution. Try again! ' )  
    end  
end
```

Ready



## 예제 3.7(3.6) [2/3]

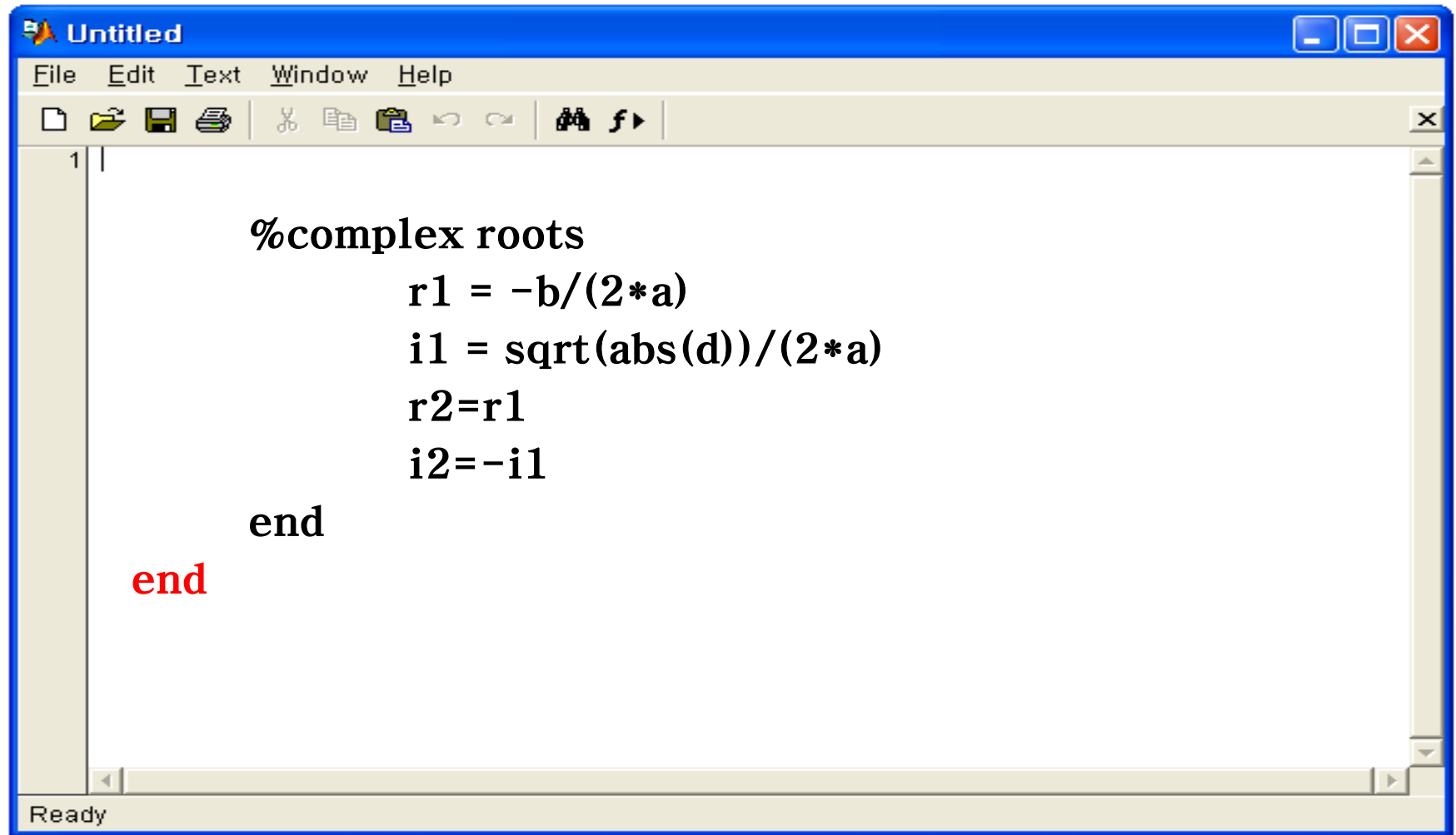


```
1 |
else
    %quadratic formula
    d=b^2 - 4*a*c; %discriminant
    if d>=0
        % real roots
        r1 = (-b+sqrt(d))/(2*a)
        r2 = (-b-sqrt(d))/(2*a)
    else
```

The image shows a MATLAB 'Untitled' window with a blue title bar and standard menu bar (File, Edit, Text, Window, Help). The toolbar includes icons for file operations and editing. The code is written in a light beige editor area with a vertical line at column 1. The code implements the quadratic formula, calculating the discriminant  $d = b^2 - 4ac$  and then the roots  $r1$  and  $r2$  based on whether  $d$  is non-negative. The status bar at the bottom indicates 'Ready'.



## 예제 3.7(3.6) (2/3)



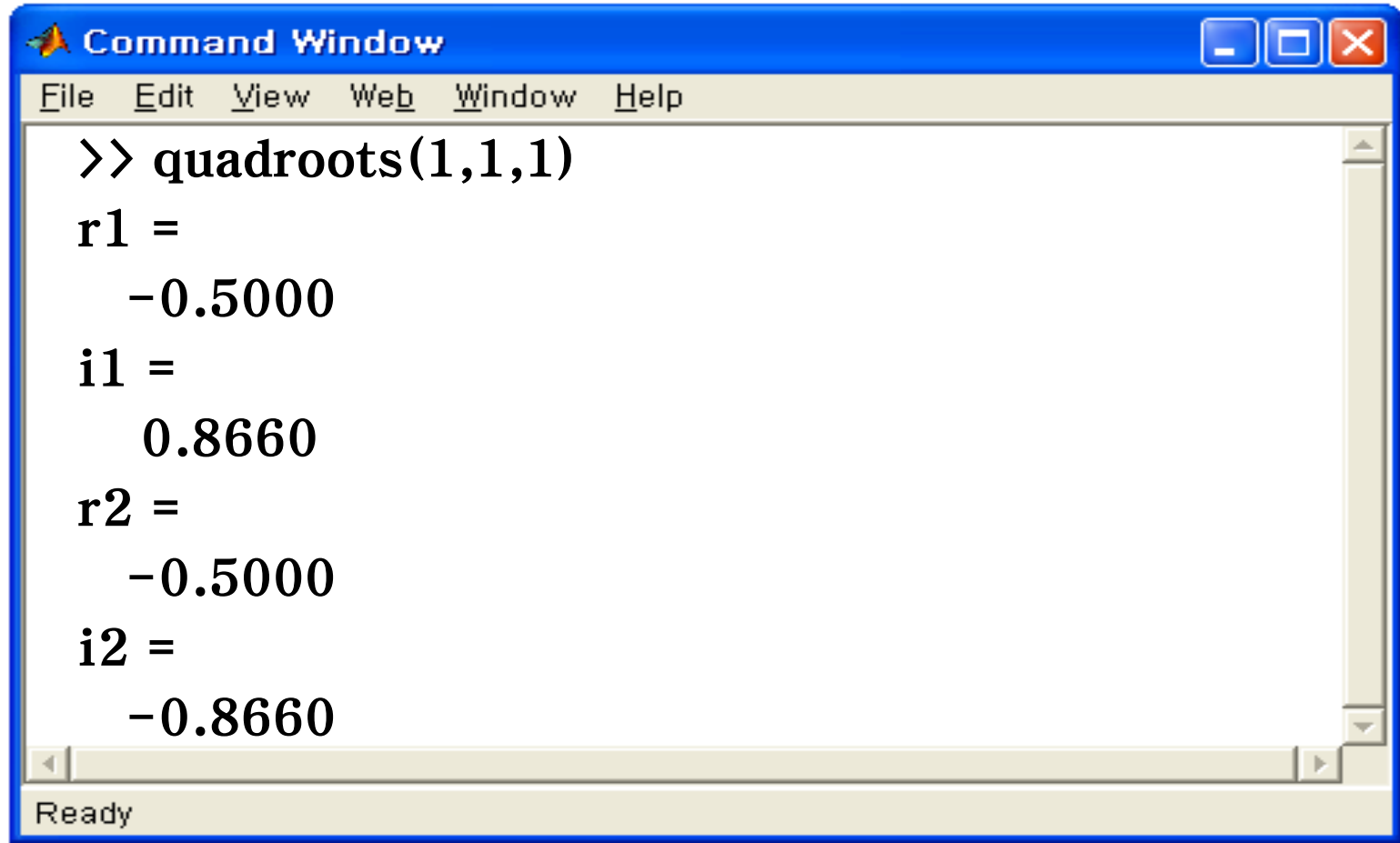
A screenshot of a MATLAB 'Untitled' window. The window has a blue title bar with the text 'Untitled' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Text', 'Window', and 'Help'. Underneath the menu bar is a toolbar with icons for file operations (new, open, save, print), editing (cut, copy, paste), and execution (run, stop). The main area of the window is a text editor with a light beige background. It contains the following MATLAB code: a line number '1' in the left margin, followed by a comment '%complex roots', then four lines of code: 'r1 = -b/(2\*a)', 'i1 = sqrt(abs(d))/(2\*a)', 'r2=r1', and 'i2=-i1'. After these, there is an 'end' statement, and finally a red 'end' statement at the bottom. The status bar at the bottom of the window shows the word 'Ready'.

```
1 |  
  
    %complex roots  
    r1 = -b/(2*a)  
    i1 = sqrt(abs(d))/(2*a)  
    r2=r1  
    i2=-i1  
  
end  
  
end
```



## 예제 3.7(3.6) [3/3]

명령창에서 다음과 같이 확인할 수 있다.

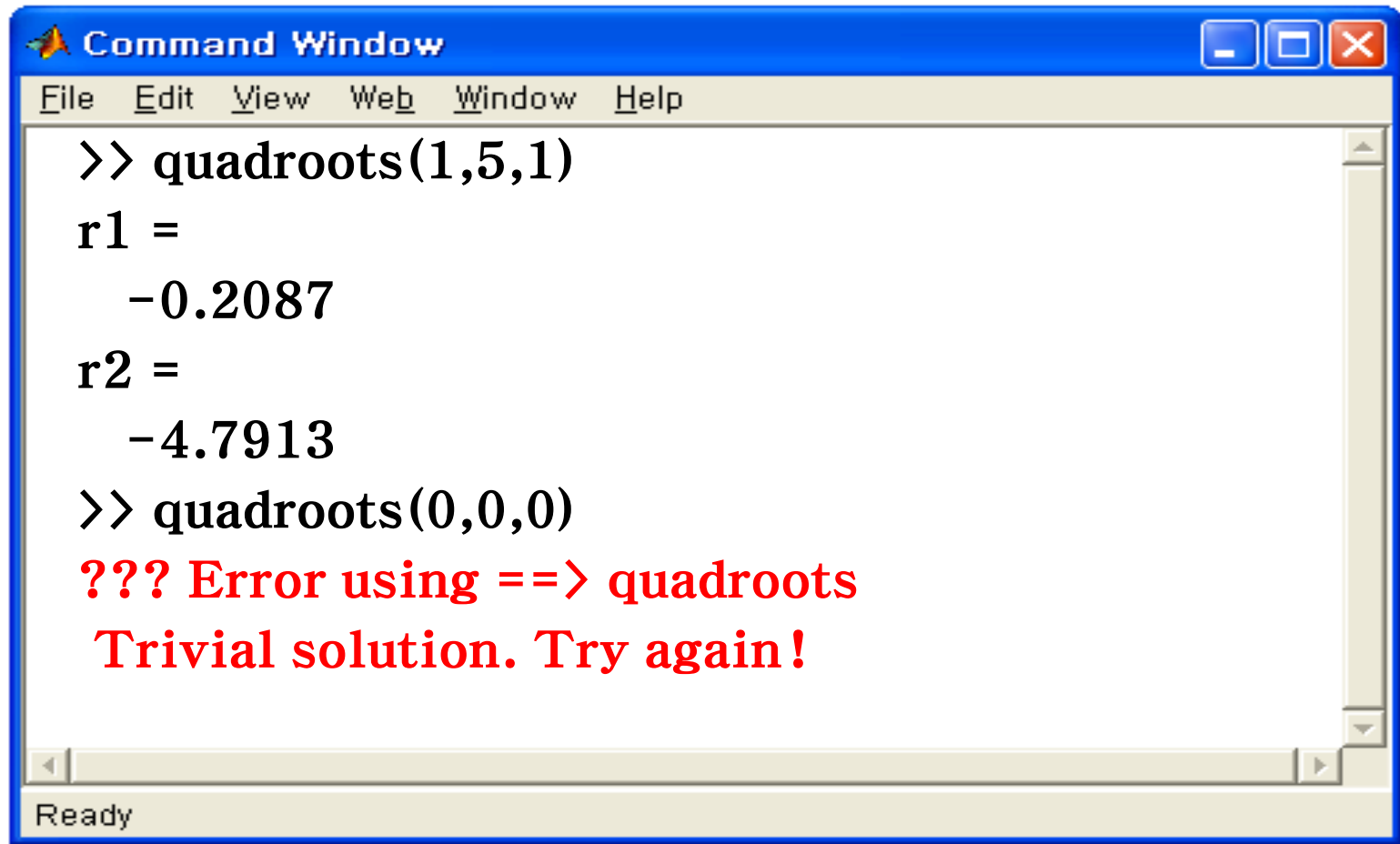
A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window contains the following text:

```
>> quadroots(1,1,1)
r1 =
    -0.5000
i1 =
    0.8660
r2 =
    -0.5000
i2 =
    -0.8660
```

At the bottom of the window, there is a status bar that says "Ready".

## 예제 3.7(3.6) [3/3]

명령창에서 다음과 같이 확인할 수 있다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text 'Command Window' and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main area of the window contains the following text:

```
>> quadroots(1,5,1)
r1 =
    -0.2087
r2 =
    -4.7913
>> quadroots(0,0,0)
??? Error using ==> quadroots
Trivial solution. Try again!
```

At the bottom of the window, there is a status bar that says 'Ready'. The window also has a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

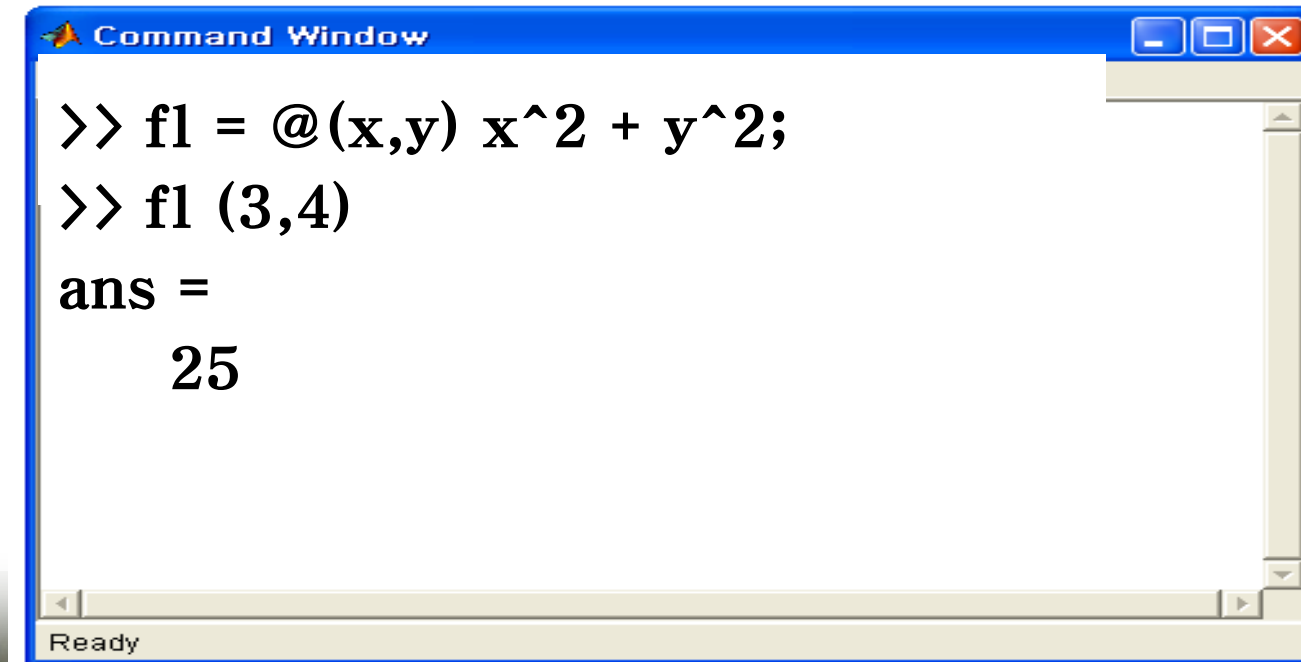


## 3.5 M-파일로의 함수 전달 (1/7)

### ■ 무명 함수

- M-파일을 만들지 않고 간단한 함수를 생성할 수 있게 한다.  
명령창에서 다음과 같은 구문을 사용한다.

```
fhandle = @(arglist) expression
```

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). The main area is white and contains the following text: ">> f1 = @(x,y) x^2 + y^2;", ">> f1 (3,4)", "ans =", and "25". The status bar at the bottom is light gray and says "Ready".

```
>> f1 = @(x,y) x^2 + y^2;  
>> f1 (3,4)  
ans =  
    25  
Ready
```

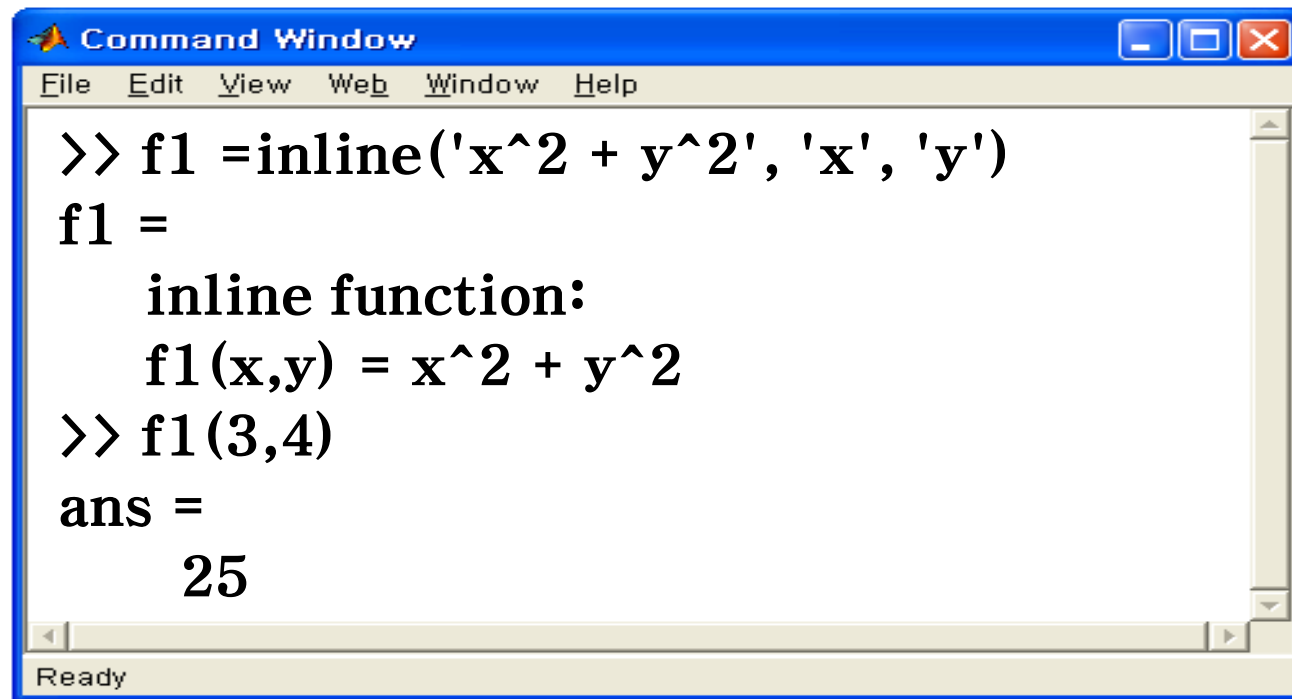


## 3.5 M-파일로의 함수 전달 (2/7)

### ■ inline 함수

– Matlab 7 이전에서 무명함수와 같은 역할 수행.

**funcname = inline('expression', 'var1' , 'var2' ,...)**



```
Command Window
File Edit View Web Window Help

>> f1 =inline('x^2 + y^2', 'x', 'y')
f1 =
    inline function:
    f1(x,y) = x^2 + y^2
>> f1(3,4)
ans =
    25

Ready
```

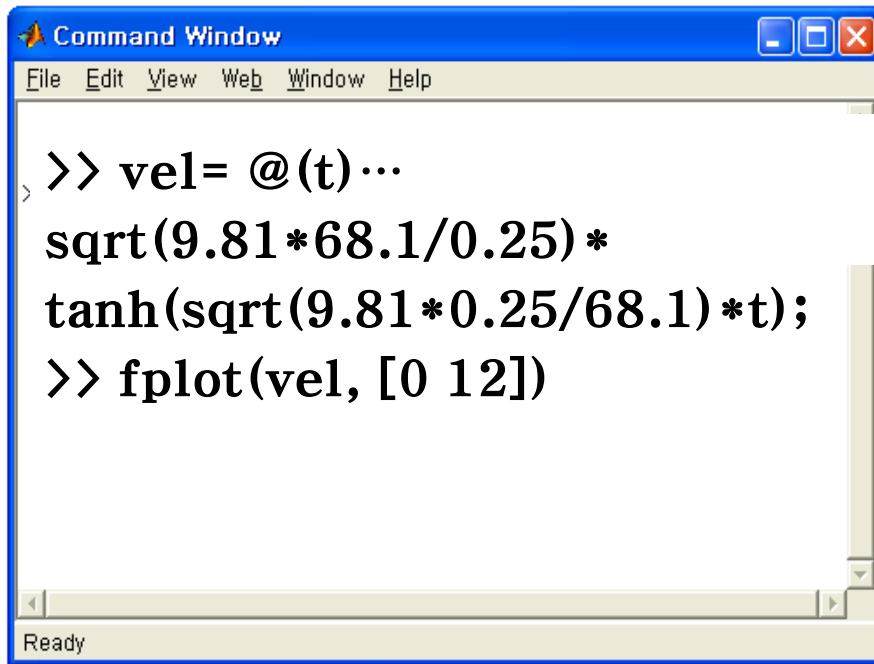


## 3.5 M-파일로의 함수 전달 (3/7)

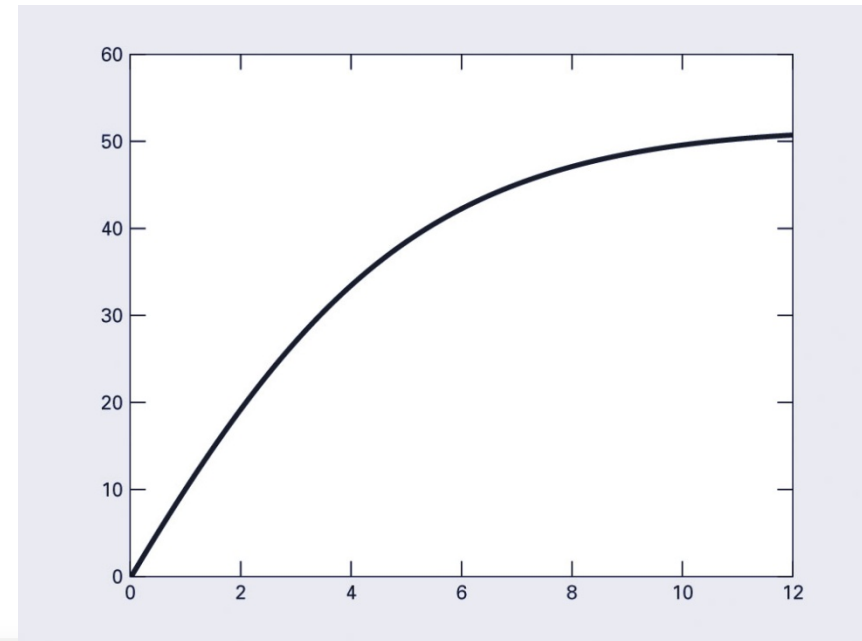
### ■ function 함수

- 다른 함수에 작동하는 함수.

예 ) `fplot (fun, lims)`

A screenshot of the MATLAB Command Window. The title bar says "Command Window". The menu bar includes "File", "Edit", "View", "Web", "Window", and "Help". The command prompt shows the following code:

```
>> vel= @(t) ...  
> sqrt(9.81*68.1/0.25)*  
tanh(sqrt(9.81*0.25/68.1)*t);  
>> fplot(vel, [0 12])
```

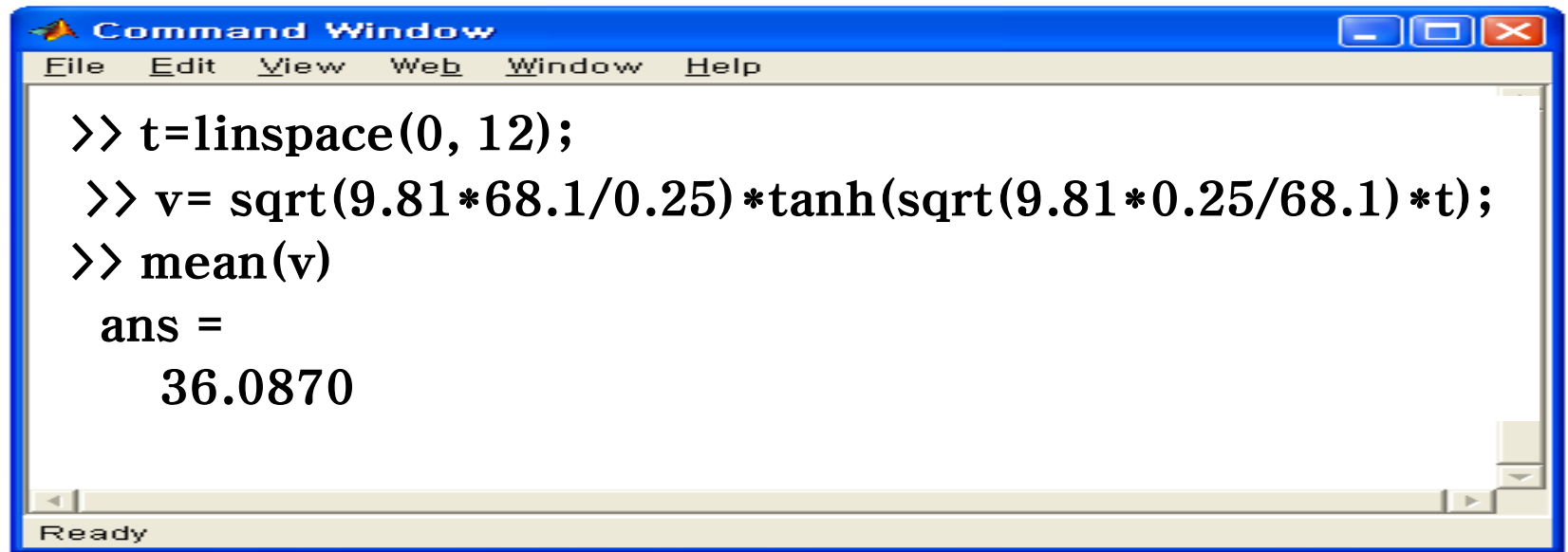
The status bar at the bottom says "Ready".

## 예제 3.8(3.7)

Q. 어떤 범위에서 함수의 평균값을 구하기 위한 M-파일 만들기

$$v(t) = \sqrt{\frac{gm}{C_d}} \tanh\left(\sqrt{\frac{gm}{C_d}} t\right)$$

풀이)  $t=0$ 에서  $t=12$ 까지의 범위에서 함수 값을 그래프로 그릴 수 있다.



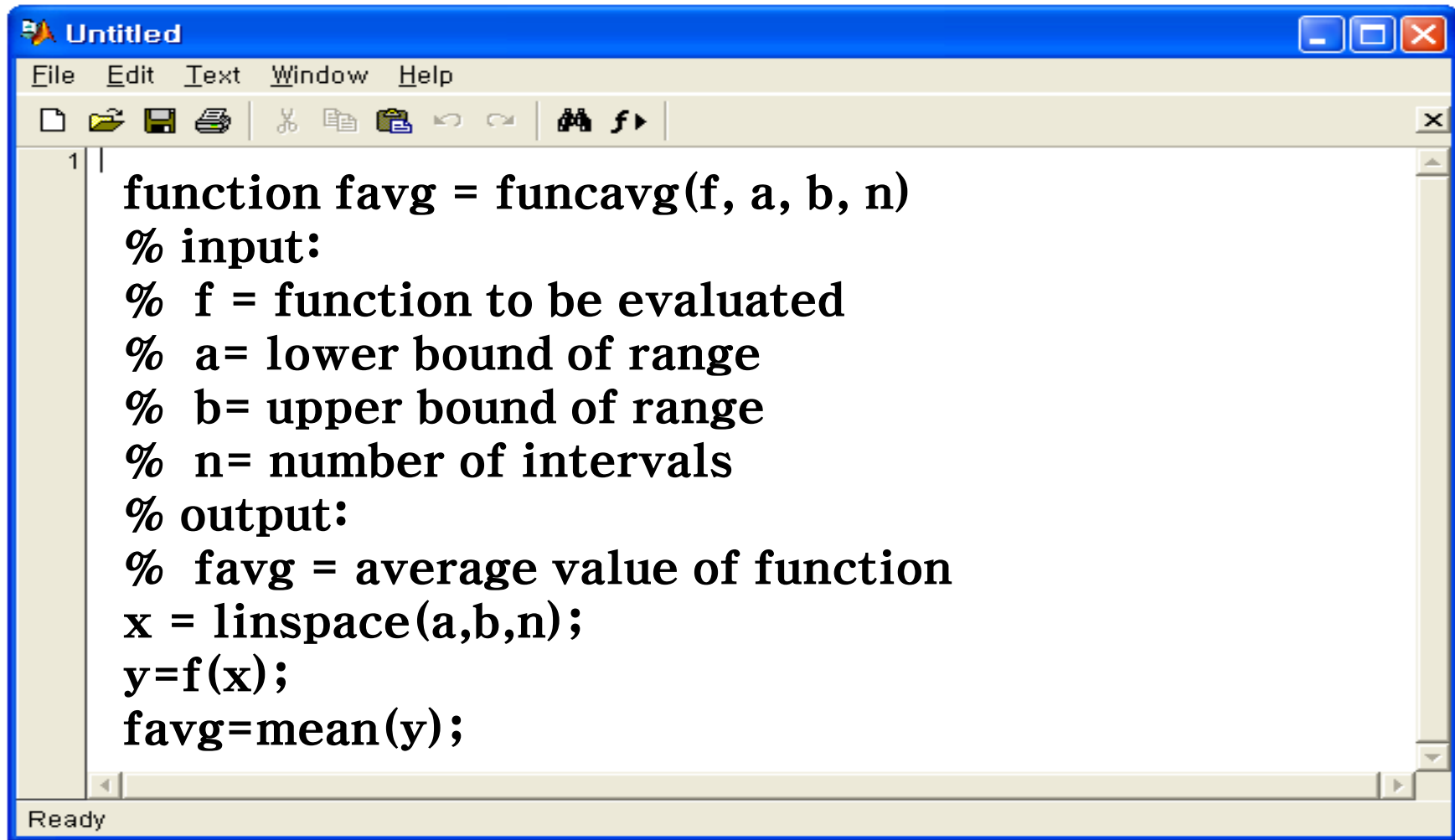
```
Command Window
File Edit View Web Window Help

>> t=linspace(0, 12);
>> v= sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
>> mean(v)
ans =
    36.0870

Ready
```



## 3.5 M-파일로의 함수 전달 (4/7)



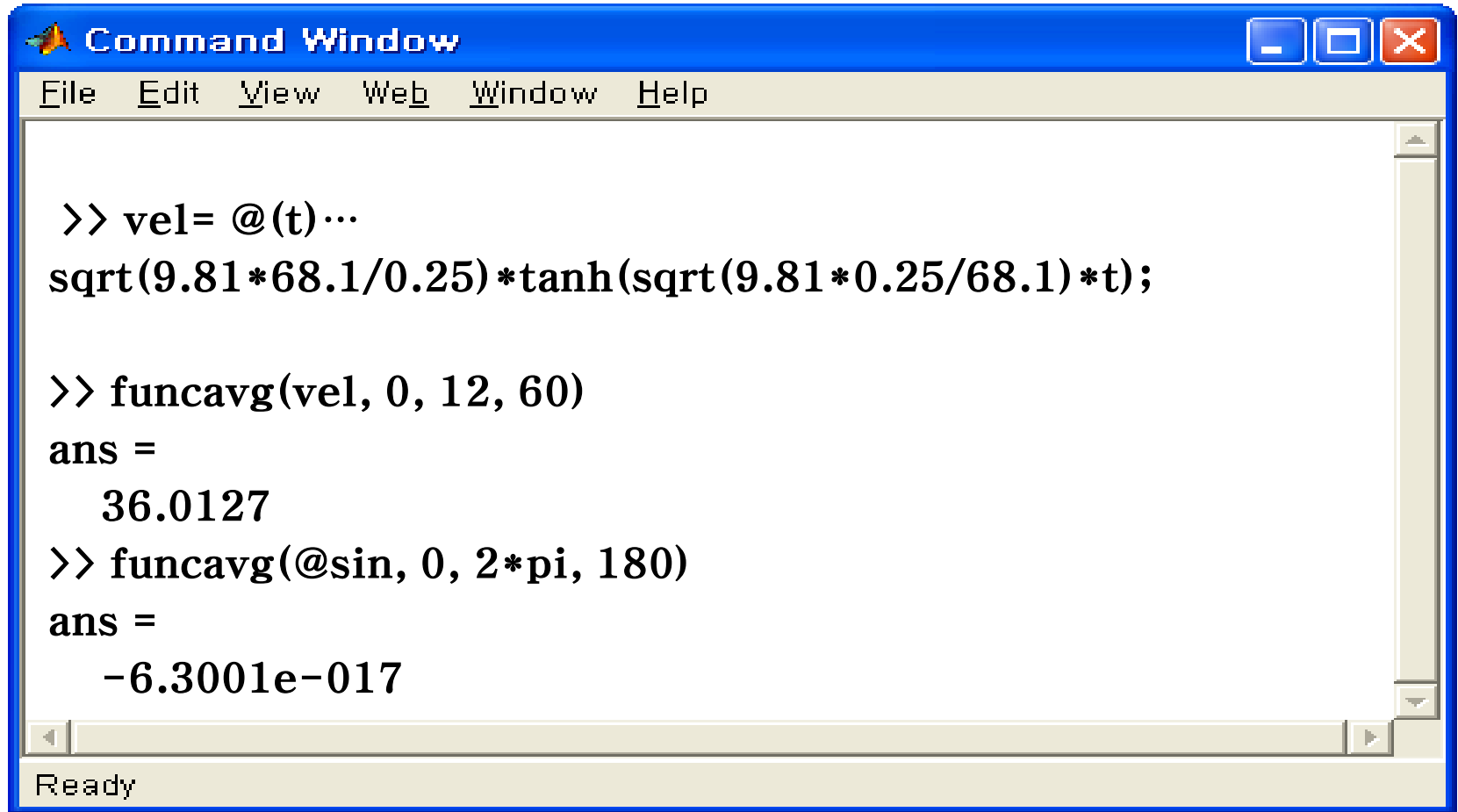
A screenshot of a MATLAB window titled "Untitled". The window has a menu bar with "File", "Edit", "Text", "Window", and "Help". Below the menu bar is a toolbar with various icons for file operations and editing. The main text area contains a MATLAB function definition for `favg`. The function takes four inputs: `f` (a function handle), `a` (lower bound), `b` (upper bound), and `n` (number of intervals). It calculates the average value of the function `f` over the range `[a, b]` using `linspace` to generate points and `mean` to average the function values. The status bar at the bottom indicates "Ready".

```
1 function favg = funcavg(f, a, b, n)
% input:
% f = function to be evaluated
% a= lower bound of range
% b= upper bound of range
% n= number of intervals
% output:
% favg = average value of function
x = linspace(a,b,n);
y=f(x);
favg=mean(y);
```



## 3.5 M-파일로의 함수 전달 (5/7)

명령창에서 다음과 같이 확인할 수 있다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window is white and contains MATLAB commands and their outputs. The commands are: 1. A function handle definition: >> vel = @(t)... sqrt(9.81\*68.1/0.25)\*tanh(sqrt(9.81\*0.25/68.1)\*t); 2. A function call: >> funcavg(vel, 0, 12, 60) followed by the output ans = 36.0127. 3. Another function call: >> funcavg(@sin, 0, 2\*pi, 180) followed by the output ans = -6.3001e-017. At the bottom of the window is a status bar that says "Ready".

```
Command Window

File Edit View Web Window Help

>> vel = @(t)...
sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);

>> funcavg(vel, 0, 12, 60)
ans =
    36.0127

>> funcavg(@sin, 0, 2*pi, 180)
ans =
   -6.3001e-017

Ready
```

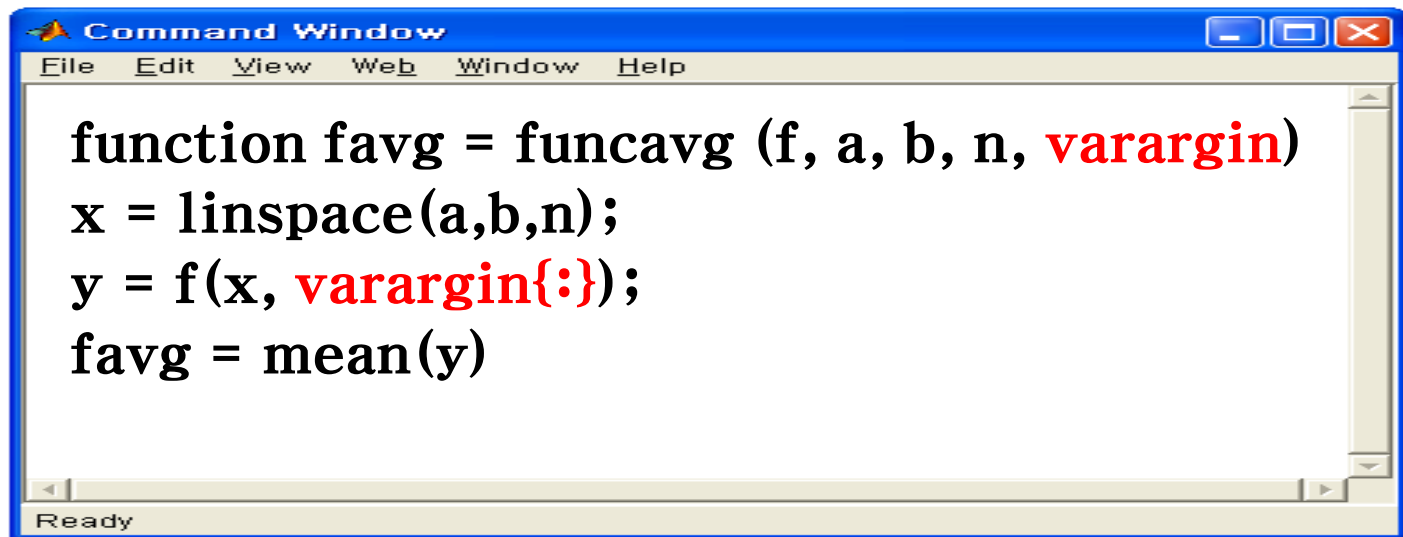


## 3.5 M-파일로의 함수 전달 (6/7)

### ■ 매개변수의 전달

- 매개변수에 새로운 값을 취할 때 편리함.
- function 함수의 마지막 입력인수에 varargin 추가함.

[ funcavg의 수정 ]



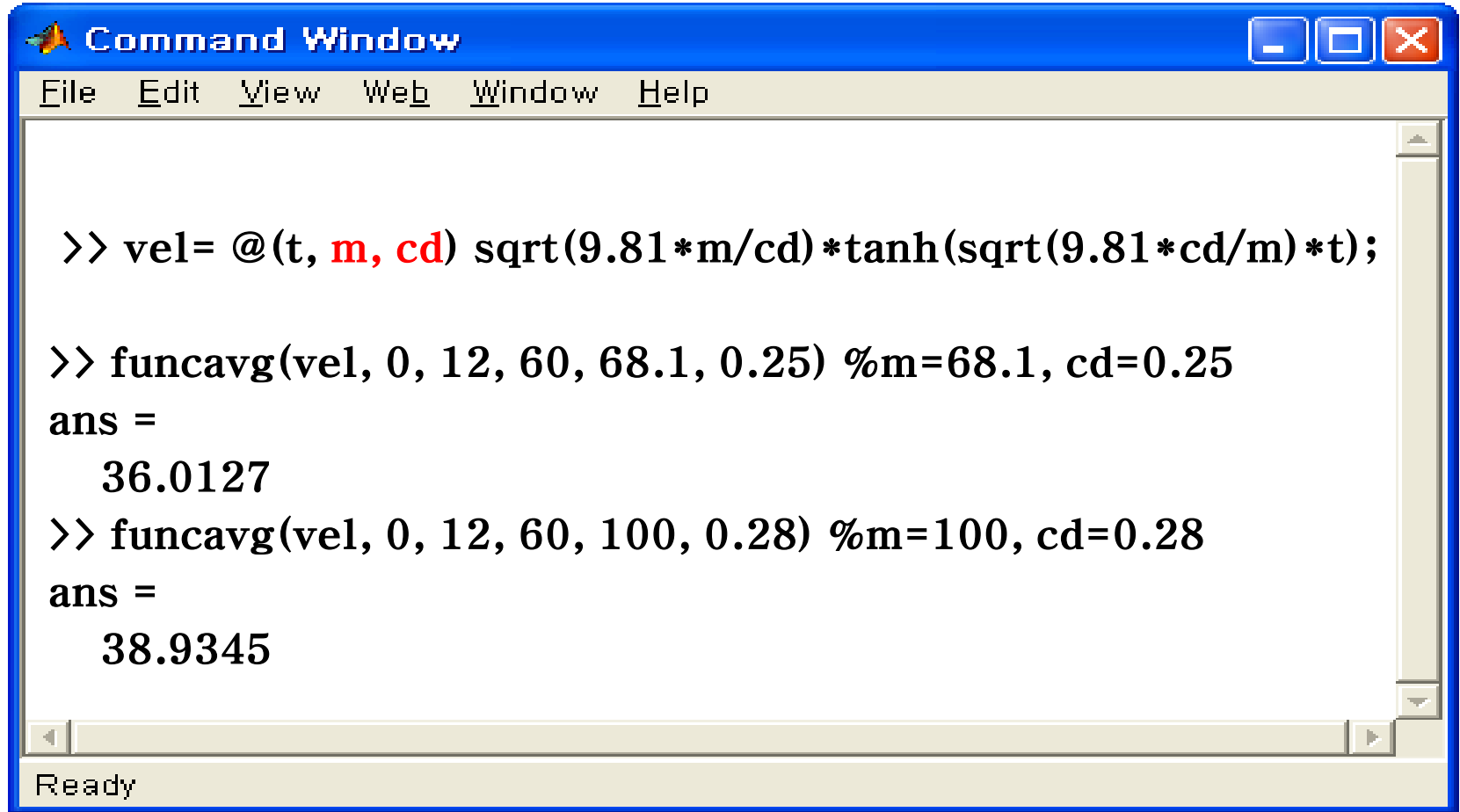
```
function favg = funcavg (f, a, b, n, varargin)
x = linspace(a,b,n);
y = f(x, varargin{:});
favg = mean(y)
```

The image shows a screenshot of the MATLAB Command Window. The window has a blue title bar with the text 'Command Window' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main area of the window contains the following MATLAB code: `function favg = funcavg (f, a, b, n, varargin)`, `x = linspace(a,b,n);`, `y = f(x, varargin{:});`, and `favg = mean(y)`. The `varargin` text in the code is highlighted in red. At the bottom of the window, there is a status bar that says 'Ready'.

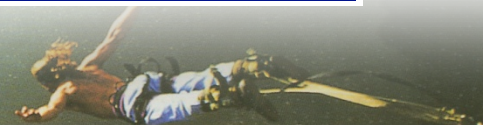


## 3.5 M-파일로의 함수 전달 (7/7)

명령창에서 다음과 같이 확인할 수 있다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window is a text editor with a white background and a vertical scrollbar on the right. It contains three lines of MATLAB code: 1. A function definition: >> vel = @(t, m, cd) sqrt(9.81\*m/cd)\*tanh(sqrt(9.81\*cd/m)\*t); 2. A function call and its output: >> funcavg(vel, 0, 12, 60, 68.1, 0.25) %m=68.1, cd=0.25  
ans =  
36.0127 3. Another function call and its output: >> funcavg(vel, 0, 12, 60, 100, 0.28) %m=100, cd=0.28  
ans =  
38.9345 At the bottom of the window is a status bar with the text "Ready".

```
>> vel = @(t, m, cd) sqrt(9.81*m/cd)*tanh(sqrt(9.81*cd/m)*t);  
  
>> funcavg(vel, 0, 12, 60, 68.1, 0.25) %m=68.1, cd=0.25  
ans =  
36.0127  
  
>> funcavg(vel, 0, 12, 60, 100, 0.28) %m=100, cd=0.28  
ans =  
38.9345  
  
Ready
```





## 3.6 사례연구: 번지점프하는 사람의 속도 (1/4)

### ■ 수학적 모델:

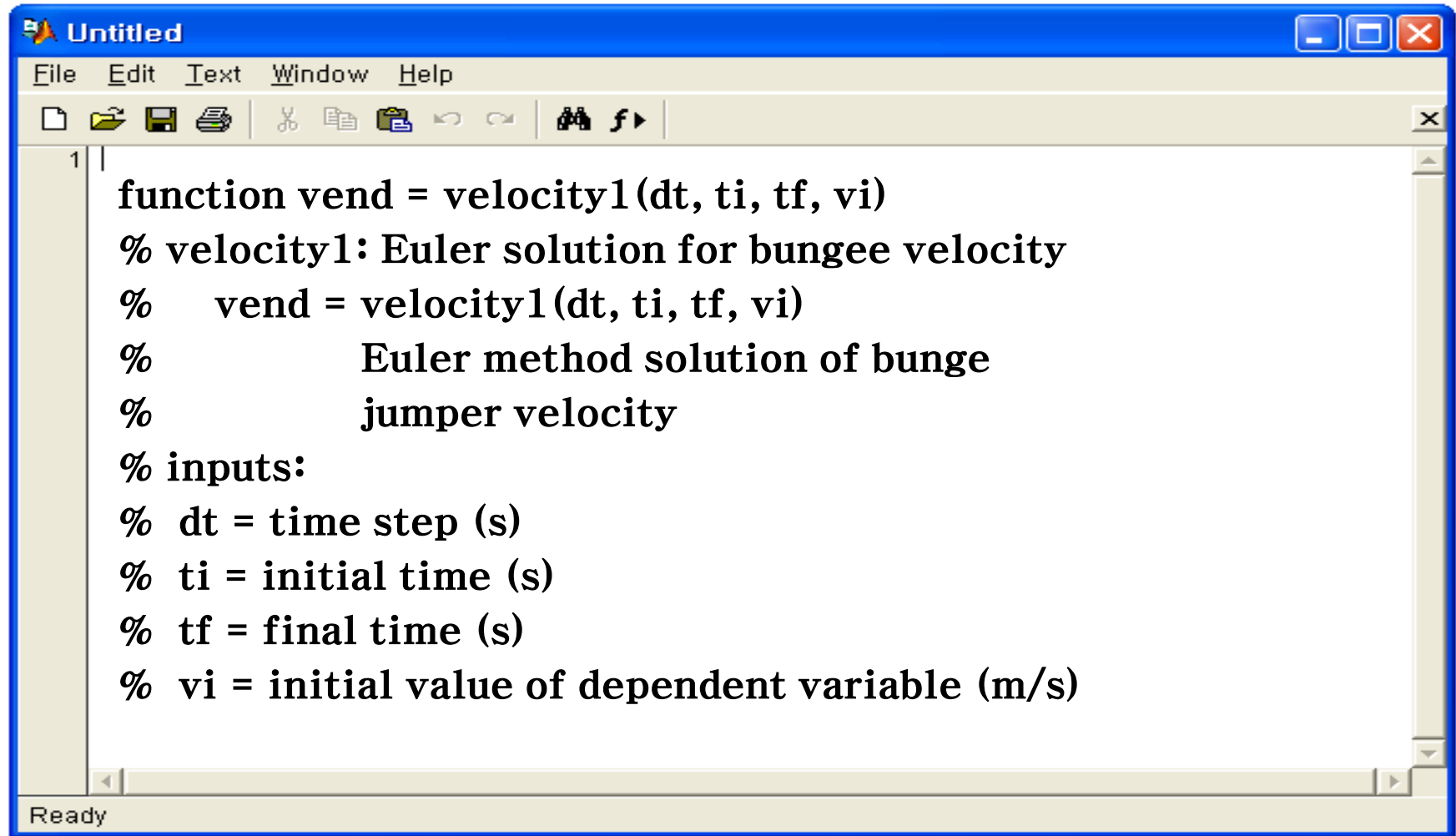
$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

### ■ Euler법:

$$v_{i+1} = v_i + \frac{dv_i}{dt} \Delta t$$



## 3.6 MATLAB M-파일: 번지점프하는 사람의 속도 (2/4)



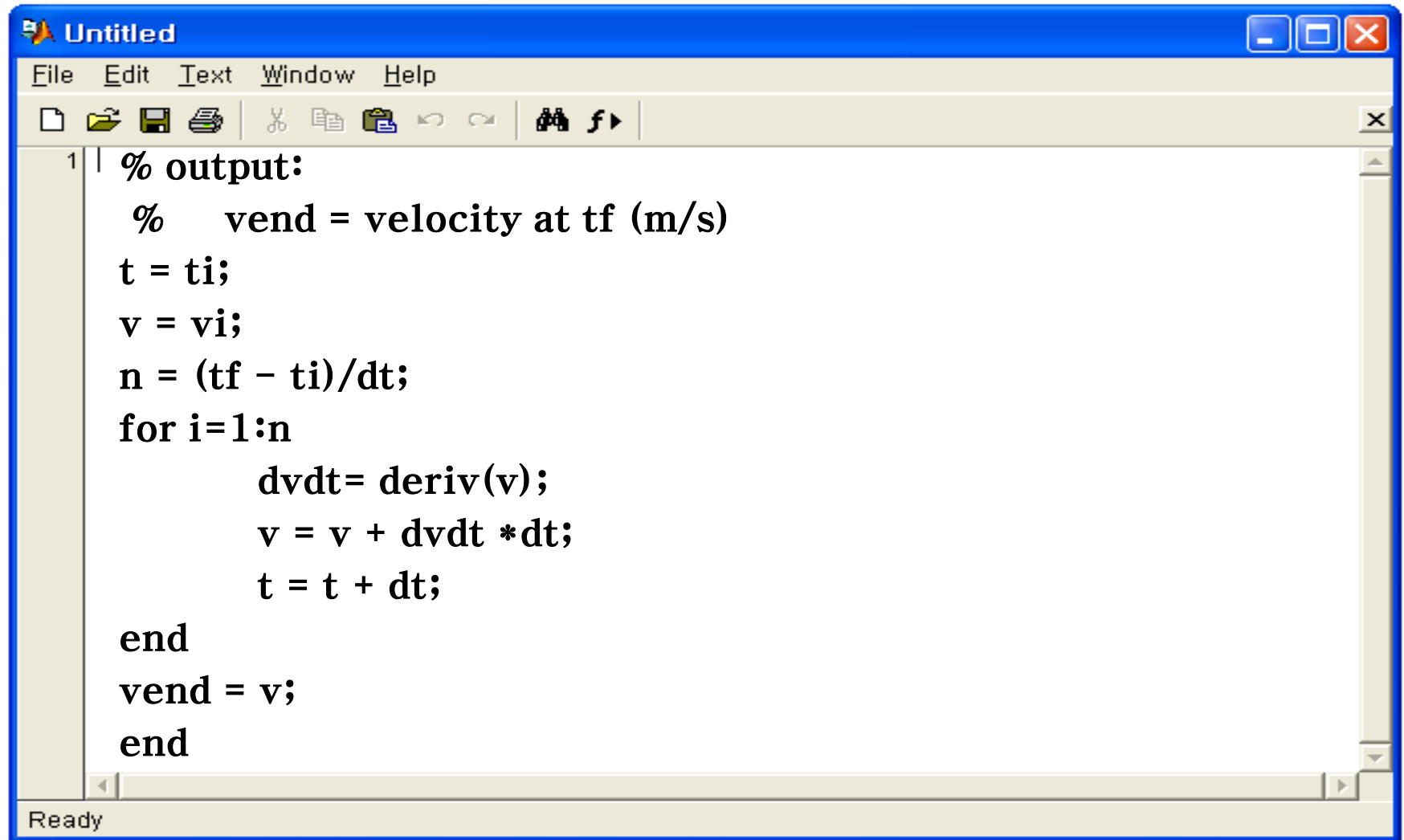
The image shows a MATLAB M-file editor window titled 'Untitled'. The window has a menu bar with 'File', 'Edit', 'Text', 'Window', and 'Help'. Below the menu bar is a toolbar with icons for file operations (new, open, save, print, copy, paste, delete) and editing (undo, redo, find, replace). The main text area contains the following MATLAB code:

```
1 | function vend = velocity1(dt, ti, tf, vi)
    % velocity1: Euler solution for bungee velocity
    % vend = velocity1(dt, ti, tf, vi)
    % Euler method solution of bungee
    % jumper velocity
    % inputs:
    % dt = time step (s)
    % ti = initial time (s)
    % tf = final time (s)
    % vi = initial value of dependent variable (m/s)
```

The status bar at the bottom of the window displays 'Ready'.



## 3.6 MATLAB M-파일: 변지점프하는 사람의 속도 (2/4)

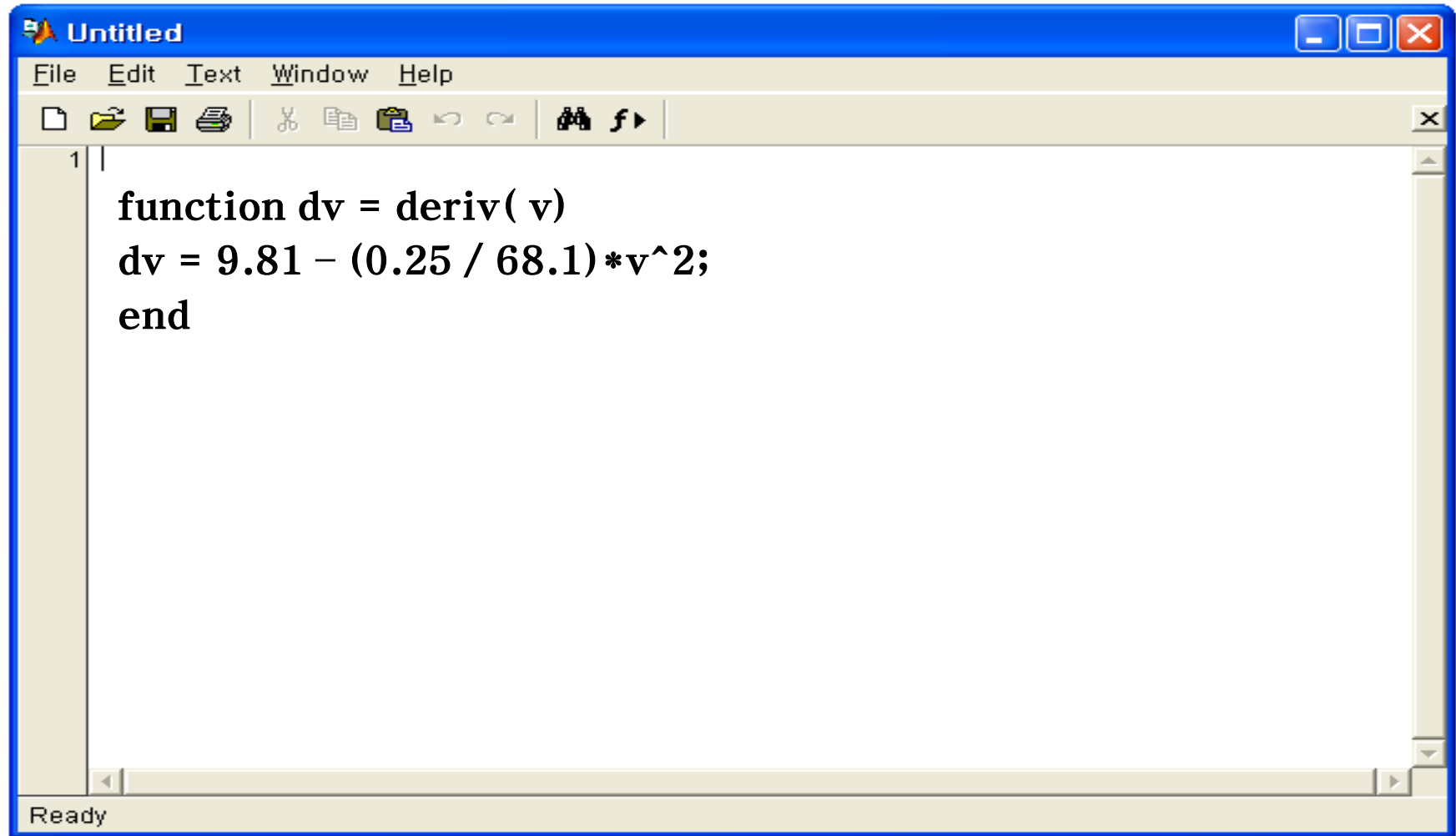


```
1 | % output:
    % vend = velocity at tf (m/s)
t = ti;
v = vi;
n = (tf - ti)/dt;
for i=1:n
    dvdt= deriv(v);
    v = v + dvdt *dt;
    t = t + dt;
end
vend = v;
end
```

Ready



## 3.6 MATLAB M-파일: 번지점프하는 사람의 속도 (3/4)



The image shows a MATLAB M-file editor window titled "Untitled". The window has a menu bar with "File", "Edit", "Text", "Window", and "Help". Below the menu bar is a toolbar with icons for file operations (new, open, save, print, copy, paste, undo, redo) and a run button. The main text area contains the following MATLAB code:

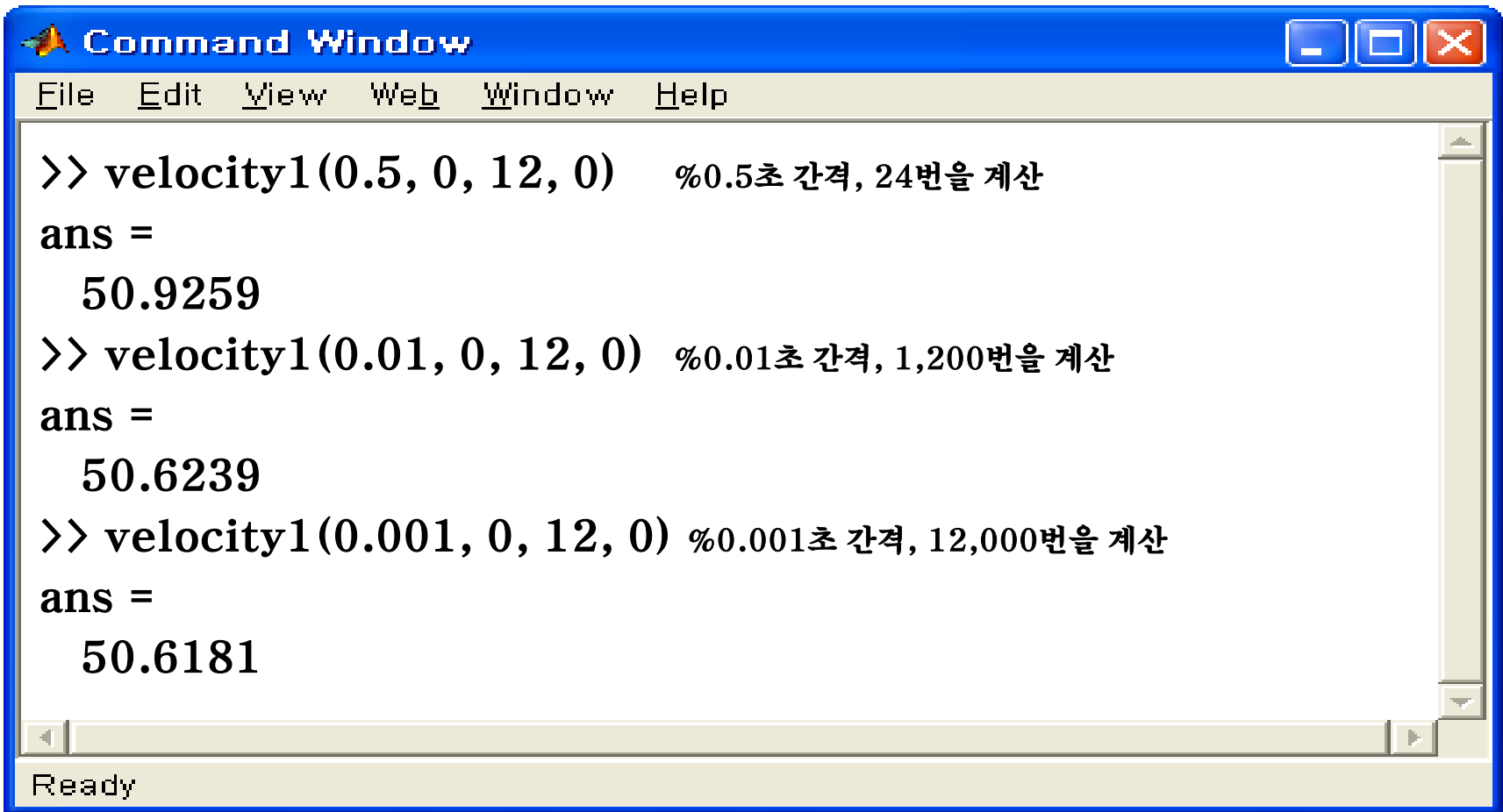
```
1 |  
function dv = deriv( v)  
dv = 9.81 - (0.25 / 68.1)*v^2;  
end
```

The status bar at the bottom of the window displays "Ready".



## 3.6 MATLAB M-파일: 번지점프하는 사람의 속도 (4/4)

명령창에서 다음과 같이 확인할 수 있다.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text 'Command Window' and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The main area of the window contains three lines of MATLAB code and their outputs. The first line is '>> velocity1(0.5, 0, 12, 0) %0.5초 간격, 24번을 계산', followed by 'ans = 50.9259'. The second line is '>> velocity1(0.01, 0, 12, 0) %0.01초 간격, 1,200번을 계산', followed by 'ans = 50.6239'. The third line is '>> velocity1(0.001, 0, 12, 0) %0.001초 간격, 12,000번을 계산', followed by 'ans = 50.6181'. At the bottom of the window is a status bar with the text 'Ready'.

```
>> velocity1(0.5, 0, 12, 0) %0.5초 간격, 24번을 계산
ans =
    50.9259
>> velocity1(0.01, 0, 12, 0) %0.01초 간격, 1,200번을 계산
ans =
    50.6239
>> velocity1(0.001, 0, 12, 0) %0.001초 간격, 12,000번을 계산
ans =
    50.6181
```

Ready



# 실습 및 과제

## ■ 실습

- 예제 3.7(3.6) 2차방정식을 풀기 위해 내포화를 이용하여 해를 구하는 함수 M-파일 만들기
- 예제 3.8(3.7) 어떤 범위에서 함수의 평균값을 구하기 위한 M-파일 만들기
- 사례연구 3.6: 번지점프하는 사람의 속도를 구하는 M-파일 만들기

■ HW: 3장연습 문제: 3.11번, 3.13번, 3.19번

