

처음 배우는 매트랩

[연습문제 공개용 답안 이용 안내]

- 본 연습문제 및 해답의 저작권은 방성완과 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136 조에 의거하여 최고 5 년 이하의 징역 또는 5 천만원 이하의 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.

Chapter 06 연습문제

6.1

```
Command Window
>> A = [2 2 9;8 9 8;3 4 7];
>> B = [1 7 7;8 4 3;9 2 4];
>> B < A
ans =
    3×3 logical array
     1     0     1
     0     1     1
     0     1     1
>> B == 4
ans =
    3×3 logical array
     0     0     0
     0     1     0
     0     0     1
```

6.2

```
Command Window
>> C = [7 7 0 3;1 -4 5 -6;5 5 1 2;3 4 2 5];
>> C(C==5) = -5
C =
     7     7     0     3
     1    -4    -5    -6
    -5    -5     1     2
     3     4     2    -5
```

6.3

```
Command Window
>> A = [16 -81 25;-9 4 36];
>> k = (A>=0);
>> A(k) = sqrt(A(k))
A =
     4    -81     5
    -9     2     6
```

6.4

```
Command Window
>> A = [2 3 -1 -3;0 0 5 9;-4 0 1 7;6 7 0 8];
>> B = [-2 3 1 0;9 0 0 -4;5 8 0 2;-7 0 6 -1];
>> A&B
ans =
4×4 logical array
     1     1     1     0
     0     0     0     1
     1     0     0     1
     1     0     0     1
>> ~A|~B
ans =
4×4 logical array
     0     0     0     1
     1     1     1     0
     0     1     1     0
     0     1     1     0
```

6.5

```
Command Window
>> C = [-2 3 1 0;9 0 0 -4;5 8 0 2;-7 0 6 -1];
>> D = [2 3 -1 -3;0 0 5 9;-4 0 1 7;6 7 0 8];
>> xor((C>=3),(D<=3))
ans =
    4×4 logical array
    1     0     1     1
    0     1     0     0
    0     0     1     0
    0     0     0     0
```

6.6

```
Command Window
>> A = [2 2 9;8 9 8;3 4 7];
>> B = [1 7 7;8 4 3;9 2 4];
>> find(A<B)
ans =
     3
     4
>> [i,j] = find(A<B)
i =
     3
     1
j =
     1
     2
```

6.7

```
Command Window
>> A = [2 3 -1 -3;0 0 5 9;-4 0 1 7;6 7 0 8];
>> [i,j] = find(A>0 & A<8, 3)
i =
     1
     4
     1
j =
     1
     1
     2
```

6.8

```
P_if1.m
1 - x = [-4,9,16,-25];
2 - if gt(x,0)
3 -     y1 = sqrt(x)
4 - else
5 -     y2 = sqrt(abs(x))
6 - end
```

입력하는 배열 **x**의 모든 원소가 '참'인 조건을 만족하지 않으므로 **if** 다음에 나오는 **y1**은 출력하지않고 다음과 같이 **else** 다음에 나오는 **y2**를 출력한다.

```
Command Window
>> P_if1
y2 =
     2     3     4     5
```

6.9

```
P_if2.m
1 - num1 = input('Enter number 1:');
2 - num2 = input('Enter number 2:');
3 - if (num1 > 0)&&(num2 == 0)
4 -     num3 = sqrt(num1)
5 - else
6 -     num3 = num1 + num2
7 - end
```

num1에 9 num2에 0을 지정하면 조건을 만족하는 참이므로 if 다음에 나오는 num3의 값을 실행하여 3이 된다. 반면에 num1에 -9를 지정하면 조건을 만족하지 않기 때문에 else 다음에 나오는 num3의 값을 실행하여 -9가 된다.

```
Command Window
>> P_If2
Enter number 1:9
Enter number 2:0
num3 =
     3

>> P_If2
Enter number 1:-9
Enter number 2:0
num3 =
    -9
```

6.10

```
P_Elseif1.m x +
1 - x = input('Enter value of x:');
2 - if isinf(x)
3 -     disp('x is an impossible value =');
4 - elseif isnan(x)
5 -     disp('x is an indeterminate value =');
6 - else
7 -     disp('x is a possible value =');
8 - end
9 - disp(x)
```

$x=1/0$ 인 불능의 값으로 입력되면 MATLAB에서 무한대를 나타내는 **Inf**(infinite)를 출력한다. $x=0/0$ 인 부정의 값으로 입력되면 MATLAB에서 숫자가 아님을 나타내는 **NaN**(not-a-number)를 출력한다. 부정과 불능이 아닌 경우에는 입력되는 숫자를 그대로 출력한다.

```
Command Window
>> P_Elseif1
Enter value of x:1/0
x is an impossible value =
    Inf
>> P_Elseif1
Enter value of x:0/0
x is an indeterminate value =
    NaN
>> P_Elseif1
Enter value of x:3
x is a possible value =
    3
```

6.11

```
P_Elseif2.m
1 - year = input('Enter the year (4 digits): ');
2 - if mod(year,400) == 0
3 -     verify = 'leap year';
4 - elseif mod(year,100) == 0
5 -     verify = 'common year';
6 - elseif mod(year,4) == 0
7 -     verify = 'leap year';
8 - else
9 -     verify = 'common year';
10 - end
11 - verify
```

2020년은 400 및 100으로 나누어 떨어지지 않기 때문에 **if**문과 첫 번째 **elseif**문을 만족하지 못하지만 4로 나누어 나머지가 0이 되므로 윤년을 확인할 수 있다. 반면에 2021년은 **if**문과 양쪽 **elseif**문의 조건을 만족하지 못하므로 평년을 확인할 수 있다.

```
Command Window
>> P_Elseif2
Enter the year (4 digits): 2020
verify =
    'leap year'
>> P_Elseif2
Enter the year (4 digits): 2021
verify =
    'common year'
```

6.12

```
P_For1.m
1 - a = 10;
2 - N = 10;
3 - x(1) = 2;
4 - for n = 1:N-1
5 -     x(n+1) = (x(n) + a/x(n))/2;
6 - end
7 - fprintf(' n           x           \n');
8 - fprintf('-----\n');
9 - fprintf('%2.0f    %17.15f\n', [1:N; x]);
```


색인 0 대신에 주어진 색인 1로 시작하는 초깃값을 사용하여 연속적으로 x 값을 10개 출력한다. 10의 제곱근 계산 결과는 여섯 째 이후 모두 동일하다.

```

Command Window
>> P_For1
n          x
-----
1    2.000000000000000
2    3.500000000000000
3    3.178571428571429
4    3.162319422150883
5    3.162277660444136
6    3.162277660168379
7    3.162277660168379
8    3.162277660168379
9    3.162277660168379
10   3.162277660168379

```

6.13

```

P_For2.m
1 - N = input('Enter N = ');
2 - for row = N:-1:2
3 -     for col = row-1:-1:1
4 -         fprintf('*');
5 -     end
6 -     fprintf('\n');
7 - end

```

내부 **for**문의 **col**의 초깃값이 **row-1**이므로 직접 입력하는 N 의 값은 5를 지정해야 첫 번째 행의 줄에서 4개의 별표를 나타낼 수 있다.

```

Command Window
>> P_For2
Enter N = 5
****
***
**
*

```

6.14

```

P_For3.m
1 - N = input('Enter N = ');
2 - for row = 2:1:N
3 -     for col = row-1:-1:1
4 -         fprintf('o');
5 -     end
6 -     fprintf('\n');
7 - end

```

내부 **for**문의 **col**의 초깃값이 **row-1**이므로 직접 입력하는 N 의 값은 5를 지정해야 네 번째 행의 줄에서 4개의 알파벳 o를 나타낼 수 있다.

```

Command Window
>> P_For3
Enter N = 5
o
oo
ooo
oooo

```

6.15

```

P_While1.m
1 - a = 10;
2 - x(1) = 2;
3 - n = 1;
4 - X = [n, x];
5 - while abs(x^2-a) > eps(a)
6 -     x = (x + a/x)/2;
7 -     n = n + 1;
8 -     X = [X; n, x];
9 - end
10 - fprintf(' n           x           \n');
11 - fprintf('-----\n');
12 - fprintf('%2.0f    %17.15f\n', X);

```

[연습문제 6.12]와 다르게 실행 횟수를 지정하지 않았기 때문에 **while**문의 조건을 만족할 때까지만 제곱근을 실행한다.

```
Command Window
>> P_While1
n          x
-----
1    2.0000000000000000
2    3.5000000000000000
3    3.178571428571429
4    3.162319422150883
5    3.162277660444136
6    3.162277660168379
```

6.16

```
P_While_if.m
1 - x = [1 -4 9 16 -25 36];
2 - m = length(x);
3 - n = 1;
4 - disp('sqrt(x) = ')
5 - while n <= m
6 -     if x(n) < 0
7 -         n = n+1;
8 -         continue
9 -     end
10 -    y = sqrt(x(n));
11 -    n = n+1;
12 -    disp(y)
13 - end
```

함수 **sqrt**를 이용하여 양수에 대한 제곱근의 계산을 실행한다.

```
Command Window
>> P_While_if
sqrt(x) =
1
3
4
6
```

6.17

```

P_For_If.m
1 - a = 10;
2 - x(1) = 2;
3 - N = 10;
4 - fprintf(' n          x(n)          #n');
5 - fprintf('-----#n');
6 - for n = 1:N-1
7 -     fprintf('%2.0f    %17.15f#n', n,x(n));
8 -     if abs(x(n)^2-a) <= eps(a)
9 -         break;
10 -    end
11 -    x(n+1) = (x(n) + a/x(n))/2;
12 - end

```

색인 0 대신에 주어진 색인 1로 시작하는 초깃값을 사용하여 연속적으로 x 값을 10개 출력한다. 10의 제곱근 계산 결과는 여섯 째 이후 모두 동일하다.

제곱근의 계산을 위해서 지정한 10회 실행 횟수와 상관없이 여섯 째 제곱근까지 if문의 조건을 만족하기 때문에 실행은 종료된다.

```

Command Window
>> P_For_If
n          x(n)
-----
1    2.000000000000000
2    3.500000000000000
3    3.178571428571429
4    3.162319422150883
5    3.162277660444136
6    3.162277660168379

```

6.18

```

P_For_If2.m
1 - format long;
2 - N = 100000;
3 - S = 1;
4 - for n = 1:N
5 -     T = (-1)^n/((2*n+1)*3^n);
6 -     if abs(T) < eps
7 -         break;
8 -     end
9 -     S = S + T;
10 - end
11 - n
12 - pai = 2*sqrt(3)*S

```

$n=30$ 이 되면 if문의 조건을 더 이상 만족하지 않고 실제 계산하려는 원주율의 값을 얻을 수 있다.

```

Command Window
>> P_For_If2
n =
    30
pai =
    3.141592653589794

```

6.19

```

P_Switch_Case1.m
1 - op = input('Select operator: ');
2 - a = input('Enter a value for a: ');
3 - b = input('Enter a value for b: ');
4 - switch (op)
5 -     case '+'
6 -         y = a + b;
7 -         disp('Addition:')
8 -     case '-'
9 -         y = a - b;
10 -        disp('Subtraction:')
11 -    case '*'
12 -        y = a*b;
13 -        disp('Multiplication:')
14 -    otherwise
15 -        y = a/b;
16 -        disp('Division:')
17 - end
18 - disp(y)

```

덧셈 연산자 '+'에 의해서 입력되는 2개 정숫값은 서로 더한 결과를 보여준다. 덧셈, 뺄셈, 곱셈을 제외한 나눗셈은 직접 연산자의 입력 없는 여백 ' '에 의해서 2개 정숫값의 나눗셈을 실행한다.

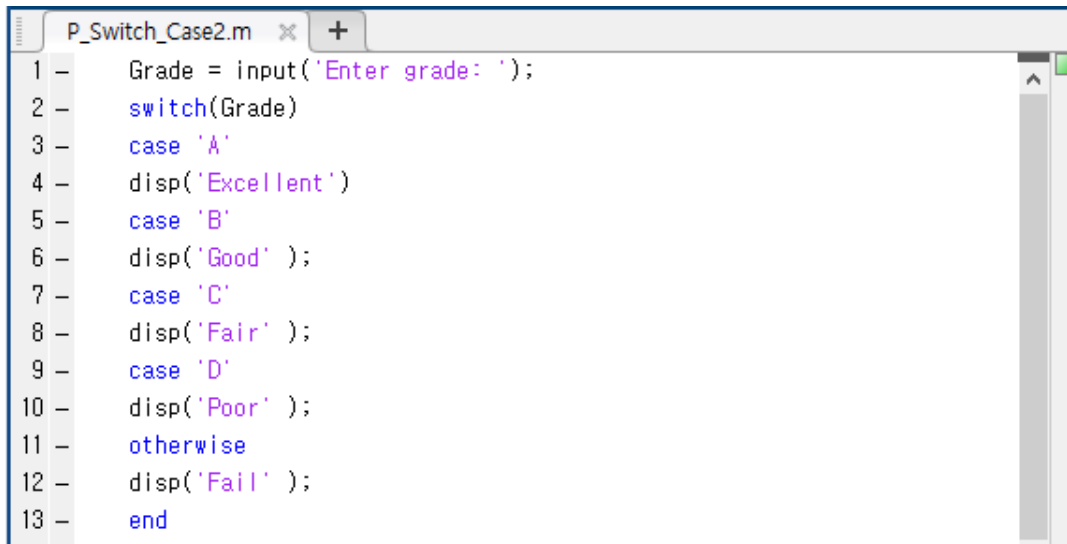
```

Command Window
>> P_Switch_Case1
Enter a value for a: 4
Enter a value for b: 2
Select operator: '+'
Addition:
    6

>> P_Switch_Case1
Enter a value for a: 4
Enter a value for b: 2
Select operator: ' '
Division:
    2

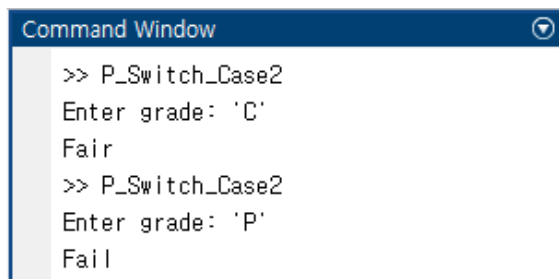
```

6.20



```
P_Switch_Case2.m
1 - Grade = input('Enter grade: ');
2 - switch(Grade)
3 - case 'A'
4 -     disp('Excellent')
5 - case 'B'
6 -     disp('Good ');
7 - case 'C'
8 -     disp('Fair ');
9 - case 'D'
10 -    disp('Poor ');
11 - otherwise
12 -    disp('Fail ');
13 - end
```

성적 'A', 'B', 'C', 'D'를 제외한 다른 알파벳이 입력되면 자동으로 'Fail'을 출력한다.



```
Command Window
>> P_Switch_Case2
Enter grade: 'C'
Fair
>> P_Switch_Case2
Enter grade: 'P'
Fail
```