

## 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 07 연습문제

### 7.1

```
Command Window
>> S = 'MATLAB is useful';
>> S1 = S(1:6)
S1 =
    'MATLAB'
>> S2 = S(8:9)
S2 =
    'is'
>> S3 = S(11:end)
S3 =
    'useful'
```

### 7.2

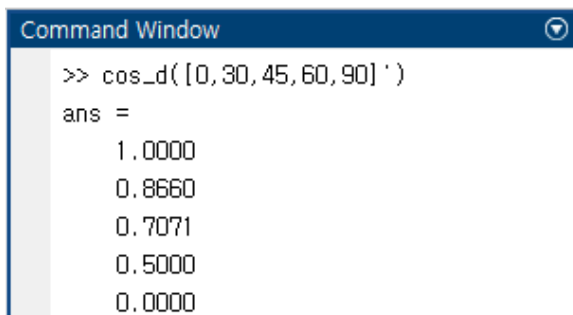
```
Command Window
>> z = 3+4*i
z =
    3.0000 + 4.0000i
>> z = 3+4*j
z =
    3.0000 + 4.0000i
>> z = 3+4i
z =
    3.0000 + 4.0000i
>> z = complex(3,4)
z =
    3.0000 + 4.0000i
>> c_z = conj(z)
c_z =
    3.0000 - 4.0000i
```

### 7.3



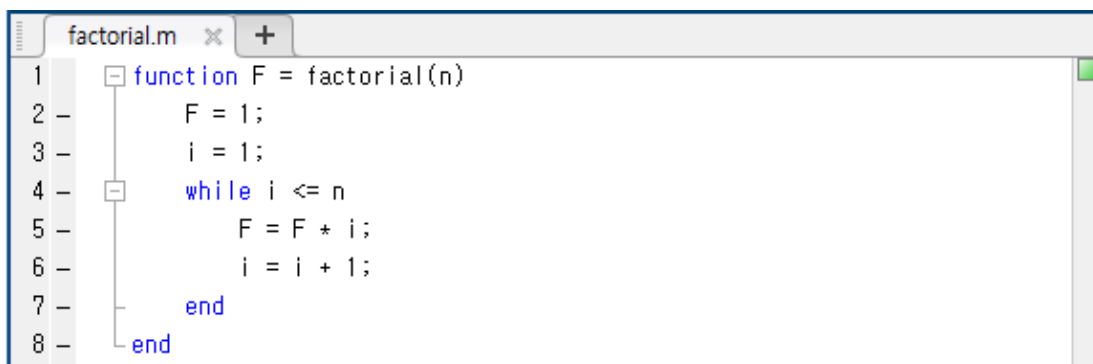
```
1 function y = cos_d(x)
2 y = cos((pi/180)*x);
```

주어진 각각의 각도를 인자로 지정하여 사용자정의함수 cos\_d 를 열벡터로 나타낸다.



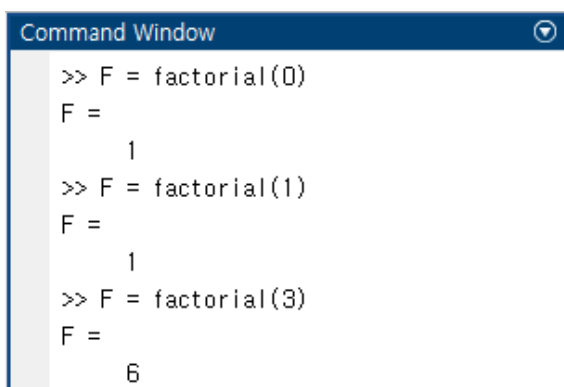
```
>> cos_d([0,30,45,60,90]')
ans =
    1.0000
    0.8660
    0.7071
    0.5000
    0.0000
```

### 7.4



```
1 function F = factorial(n)
2     F = 1;
3     i = 1;
4     while i <= n
5         F = F * i;
6         i = i + 1;
7     end
8 end
```

주어진 3 개의 값에 대한 계승을 계산하면 다음과 같다.



```
>> F = factorial(0)
F =
    1
>> F = factorial(1)
F =
    1
>> F = factorial(3)
F =
    6
```

## 7.5

```
1  function F = fibo(n)
2  -    if (n <= 1)
3  -        F = n;
4  -    else
5  -        F = fibo(n - 1) + fibo(n - 2);
6  -    end
```

처음 3 개 항과 열 번째 항에 대한 피보나치 수열을 실행하면 다음과 같다.

```
Command Window
>> F = fibo(0)
F =
    0
>> F = fibo(1)
F =
    1
>> F = fibo(2)
F =
    1
>> F = fibo(10)
F =
   55
```

## 7.6

```
1  function C = bicoef(n,k)
2  -    if (k == 0) || (k == n)
3  -        C = 1;
4  -    elseif k > n
5  -        C = 0;
6  -    else
7  -        C = bicoef(n-1,k) + bicoef(n-1,k-1);
8  -    end
```

다양한 정수  $n$  과  $k$  을 지정하여 이항 계수를 계산한 결과는 다음과 같다.

```
Command Window
>> C = bicoef(0,0)
C =
    1
>> C = bicoef(2,4)
C =
    0
>> C = bicoef(4,2)
C =
    6
```

## 7.7

```
Pastri.m
1 - for n = 0:7
2 -     C = [];
3 -     for k = 0:n
4 -         C = [C, bicoef(n,k)];
5 -     end
6 -     disp(C);
7 - end
```

외부 **for** 문에  $n$  을 지정하고 내부 **for** 문에  $k$  를 지정하여 파스칼 삼각형을 실행한 결과는 다음과 같다.

```
Command Window
>> Pastri
    1
    1    1
    1    2    1
    1    3    3    1
    1    4    6    4    1
    1    5   10   10    5    1
    1    6   15   20   15    6    1
    1    7   21   35   35   21    7    1
```

## 7.8

```
Prms_ma.m  x  +
1  function [R,M] = Prms_ma(x)
2  -    R = rms(x);
3  -    M = ma(x);
4  -    function y = rms(x)
5  -    y = sqrt(sum(abs(x).^2) / length(x));
6  -    function y = ma(x)
7  -    y = sum(abs(x)) / length(x);
```

사용자정의함수를 실행한 결과는 다음과 같다.

```
Command Window
>> x = -5:5;
>> [R,M] = Prms_ma(x)
R =
    3.1623
M =
    2.7273
```

## 7.9

```
H_L.m  x  +
1  function [L,H] = H_L(first, second)
2  -    if first < second
3  -        L = first;
4  -        H = second;
5  -    else
6  -        L = second;
7  -        H = first;
8  -    end
9  - end
```

```
Command Window
>> [L,H] = H_L(5,8)
L =
    5
H =
    8
>> [L,H] = H_L(9,2)
L =
    2
H =
    9
```

## 7.10

```
dotprod.m
1 function D = dotprod(A,B)
2     n = length(A);
3     D = 0;
4     for i = 1:n
5         D = D+A(i)+B(i);
6     end
```

2 개의 주어진 벡터를 이용하여 벡터의 내적을 구한 결과는 스칼라 값이다.

```
Command Window
>> D = dotprod([1 2 -3],[0 -4 1])
D =
   -11
```

## 7.11

```
crossprod.m  x  +
1  function C = crossprod(A,B)
2  -     n = length(A);
3  -     if n == 2
4  -         A(3) = 0;
5  -         B(3) = 0;
6  -     end
7  -     C(1) = A(2)*B(3)-A(3)*B(2);
8  -     C(2) = A(3)*B(1)-A(1)*B(3);
9  -     C(3) = A(1)*B(2)-A(2)*B(1);
```

2 개의 주어진 벡터를 이용하여 벡터의 외적을 구한 결과는 벡터이다.

```
Command Window
>> C = crossprod([1 2 -3],[0 -4 1])
C =
    -10     -1     -4
```

## 7.12

```
exchange.m  x  ex_rate.m  x  +
1  function [D, Y] = exchange(won, dollarPwon, yenPwon)
2  -     D = won*dollarPwon;
3  -     Y = won*yenPwon;
4  - end
```

원화에 1 원당 달러의 환율을 곱하면 전체 달러를 얻고, 원화에 1 원당 엔의 환율을 곱하면 전체 엔을 얻을 수 있다.

```
ex_rate.m  x  +
1  -     won = input('Total amount of Korean Won: ');
2  -     dollarPwon = 1/1204;
3  -     yenPwon = 1/10.79;
4  -     [D, Y] = exchange(won, dollarPwon, yenPwon);
5  -     fprintf('This is %6.2f dollar \n', D);
6  -     fprintf('This is %6.0f yen \n', Y);
```



한화 100,000 원을 입력하여 환율이 계산된 미화 및 엔화를 얻을 수 있다.

```
Command Window
>> ex_rate
Total amount of Korean Won: 100000
This is 83.06 dollar
This is 9268 yen
```

### 7.13

```
dworder.m
1 function y = dworder(x)
2     n = length(x);
3     for i = 1:n-1
4         T = x(i);
5         for j = i+1:n
6             if x(j) > T
7                 x(i) = x(j);
8                 x(j) = T;
9                 T = x(i);
10            end
11        end
12    end
13    y = x;
```

주어진 배열의 원소중 가장 큰 88 부터 가장 작은 15 까지 내림차순으로 재배열하여 나타낸다.

```
Command Window
>> x = [51 51 42 39 55 71 45 62 88 85 15 73 78 34 69];
>> y = dworder(x)
y =
Columns 1 through 13
    88    85    78    73    71    69    62    55    51    51    45    42    39
Columns 14 through 15
    34    15
```

## 7.14

```

1  function [r,arg]= recTpol(x,y)
2  -   r = sqrt(x^2+y^2);
3  -   arg = acos(abs(x)/r)+180/pi;
4  -   if (x < 0) && (y > 0)
5  -       arg = 180 - arg;
6  -   elseif (x < 0) && (y < 0)
7  -       arg = 180 + arg;
8  -   elseif (x > 0) && (y < 0)
9  -       arg = -arg;
10 -   else
11 -   end

```

각각의 사분면에 위치한 4 개의 복소수  $z=3+j4$ ,  $z=-3+j4$ ,  $z=-3-j4$ ,  $z=3-j4$ 를 실행하여 얻은 극 형식의 크기 및 편각을 실행한 결과를 보여준다.

```

Command Window
>> [r,arg] = recTpol(3,4)
r =
    5
arg =
   53.1301
>> [r,arg] = recTpol(-3,4)
r =
    5
arg =
  126.8699
>> [r,arg] = recTpol(-3,-4)
r =
    5
arg =
  233.1301
>> [r,arg] = recTpol(3,-4)
r =
    5
arg =
  -53.1301

```

## 7.15

```

1  function nroot(c, n, x0, e)
2
3  -   x0 = x0;
4
5  -   j = 0;    % to count the number of iterations
6
7  -   xn = x0 - (c-x0^n)/ (-n*(x0^(n-1))); % Newton's method
8
9  -   fprintf(' %t      xn %t      iterations ');
10 -   fprintf('%n ----- %t %t -----');
11
12 -   while (abs(xn - x0)) > e
13 -       x0 = xn;
14 -       xn = x0 - (c-x0^n)/ (-n*(x0^(n-1)));
15 -       j = j+1;
16 -   end
17 -   j = j+1;
18 -   fprintf('%n %20.20f      %9d%n', xn, j);

```

6 회 반복으로 허용 오차  $10^{-6}$ 를 만족하는  $\sqrt{8}$ 의 근의 값이 계산되고, 반면에  $\sqrt[3]{8}$ 의 근은 7 회 반복으로 얻을 수 있다.

```

Command Window
>> nroot(8,2,1,10^-6)
      xn      iterations
-----
2.82842712474619029095      6
>> nroot(8,3,1,10^-6)
      xn      iterations
-----
2.00000000000000000000      7

```