1. compute the number of shortest paths from (0,0) to (5,2) by factorial function.

   A. f(5,2) = f(5,1)+f(4,2) = f(5,0)+f(4,1)+f(4,1)+f(3,2) =
   1 + 2(f(4,0) + f(3,1)) + f(3,1) + f(2,2) =
   3 + 3(f(3,0) + f(2,1)) + f(2,1) + f(1,2) =
   6 + 4(f(2,0) + f(1,1)) + f(1,1) + f(0,2) =
   11 + 5(f(1,0) + f(0,1)) = 21

2. by binomial coefficient?
   A. f(5,2) = 7C2 = 21

3. nonsingular?
   A. A is nonsingular matrix iff Ax = 0인 x가 only x=0
   Ax=0 if x=0

4. solution type of linear system of equation(Ax = b)
   (1) nonsingular, homogeneous : one unique solution which is 0 -> trivial solution
   (2) singular, homogeneous : infinitely many solution, include 0 -> nontrivial solution
   (3) nonsingular, nonhomogeneous : one unique solution, which is nonzero
   (4) singular, nonhomogeneous(with solution) : infinitely many solution, b ∈ range of A
   -> particular solution + general solution(z, z∈N(A))
   (5) singular, nonhomogeneous(no solution) :

$b \notin$ range of A

5. 다음은 n-set의 j-subsets 들을 generating 하는 matlab code이다. 질문에 답하여라.

```
function b = general(n,j)
b = 1:j;
disp(b)
while true
    h = j+1;
    found = false;
    while h > 1 && ~found
        h = h-1;
        if b(h) < n + h - j
            found = true;
        end
    end
    if ~found
        break
    end
    b(h) = b(h)+1;
    for k = h+1 : j
        b(k) = b(k-1) + 1;
    end
    disp(b)
end
end
```

Q1. h는 무엇을 의미하는가?
Q2. found는 언제 true가 되는가? 의미하는 것은?
Q3. for문은 언제 실행되는가? for문이 실행되기 전과 후의 결과를 예시를 통해서 설명하시오.

A1. h는 이번에 올림을 할 자리수를 의미한다.
예를 들어 n,j가 각각 5,3이고 현재 b = [1,3,4]이면
다음은 [1,3,5]가 될수있으므로 h = 3이다.
만약 [1,3,5]라면 5는 이제 올릴수 없으므로 h = 2이다.

A2. found는 b(h) < n + h - j일 때 true가 된다.
b의 h번째 자리수에서 가능한 값은 h <= b(h) <= n-j+h이다. 따라서 b(h) 값이 n-j+h보다 작다면 그 자리는 아직 올릴 수 있으므로 found true가 된다.

A3. for문은 자릿수 올림이 일어났을 때 실행된다. 자리 올림이 일어나지 않았다면 h = j이었을 것이고, 이때 for문은 실행되지 않는다. 예를 들어서 b가 [1,4,5]였다면 이번에 자릿수 올림을 할 자리는 1이고, 자릿수를 더한후 [2,4,5]가 된다. 하지만 실제 [1,4,5] 다음수는 [2,3,4]이다. 이를 맞추기 위해 이번에 자릿수 올림을 진행한 다음 숫자부터 값을 이전 자릿수 + 1로 설정해준다.
[1,2,5] 였다면 [1,3,4]가 된다.

Q4. general(4,2)의 결과를 쓰시오.
A4. [1,2][1,3][1,4][2,3][2,4][3,4]

1. For given matrix A,

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 & 5 \\ -2 & -4 & 0 & 4 & -2 \\ 1 & 2 & 2 & 4 & 9 \end{bmatrix}$$

find the 4 fundamental subspaces of A which are $R(A)$, $N(A)$, $R(A^T)$, $N(A^T)$.

2. For given matrix B,

$$B = \begin{bmatrix} 1 & 2 & 3 & -5 & -5 \\ 3 & 6 & 1 & -7 & 9 \\ 1 & 2 & 0 & -2 & 4 \\ 0 & 0 & 1 & -1 & -3 \\ 2 & 4 & 1 & -5 & 5 \end{bmatrix}$$

find the 4 fundamental subspaces of B and justify that (1) $[R(B)]^{\perp} = N(B^T)$ and (2) $N(B) = [R(B^T)]^{\perp}$.

1. rref(A) =

1	2	0	-2	1
0	0	1	3	4
0	0	0	0	0

column1,3 are linearly independent

R(A)={[1 -2 1]'[1 0 2]'}

x1 + 2x2 -2x4 + x5 = 0

x1 = -2x2 + 2x4 - x5

x3 + 3x4 + 4x5 = 0

x3 = -3x4 -4x5

[x1 x2 x3 x4 x5]' = x2[-2 1 0 0 0]' +

x4[2 0 -3 1 0]' + x5[-1 0 -4 0 1]'

N(A)={[-2 1 0 0 0]' [2 0 -3 1 0]' [-1 0 -4 0 1]'} rref(A’) =

$$\begin{matrix} 1.0000 & 0 & 2.0000 \\ 0 & 1.0000 & 0.5000 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

column1,2 are linearly independent

R(A’) = {[1 2 1 1 5]’ [-2 -4 0 4 -2]’}

x1 + 2x3 = 0

x1 = -2x3

x2 + 0.5x3 = 0

x2 = -0.5x3

N(A’) = {[-2 -0.5 1]’}

2.

R(B) = {[1 3 1 0 2]’ [3 1 0 1 1]’}

N(B) = {[-2 1 0 0 0]’ [2 0 1 1 0]’ [-4 0 3 0 1]’}

R(B’) = {[1 2 3 -5 -5]’ [3 6 1 -7 9]’}

N(B’) = {[1/8 -3/8 1 0 0]’ [-3/8 1/8 0 1 0]’
[-1/8 -5/8 0 0 1]}

R(B) ⊥ N(B’) : all basis product = 0

R(B’) ⊥ N(B) : also

코딩하세요.

1. step(x) = 0 (x<0)

               1 otherwise

```
function y = step(x)
y = zeros(size(x));
set1 = find(x>=0);
y(set1) = ones(size(set1));
```

2. ramp(x) = 0 (x<0)

              x otherwise

```
function y = ramp(x)
y = zeros(size(x));
set1 = find(x>=0);
y(set1) = x(set1);
```

3.

g(x) = 0 (x<0)

       sin(pix/2) (0<=x<=1)

       1 otherwise

```
function y = g(x)
y = ones(size(x));
set1 = find(x<0);
set2 = find(0<= x & x<= 1);
y(set1) = zeros(size(set1));
y(set2) = sin(pi*x(set2)/2);
```

pascal triangle code

```
function r = pc(x)
for i = 1:x
    for j = 1:x
        if i==1 | j==1
            r(i,j) = 1;
        else
            r(i,j) = r(i-1,j)+r(i,j-1);
        end
    end
end
```

vander code

```
function r = vd(x)
s = size(x,2);
for i = 1:s
    for j = s:-1:1
        if j == s
            r(i,j) = 1;
        else
            r(i,j) = r(i,j+1) * x(i);
        end
    end
end
```