0행 0열부터 시작 아닌가?

- 행과 열
  - m개의 행과 n개의 열을 가진 행렬

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

$$\begin{pmatrix} 3 & 4 & 1 \\ 2 & 7 & 4 \\ 5 & 8 & 9 \end{pmatrix} \quad a_{12} = ? \qquad \begin{pmatrix} 6 & 3 \\ 3 & 9 \\ 2 & 1 \end{pmatrix} = \underline{3}\text{행 } \underline{2}\text{열}$$

행렬 클래스

```
class mat3 {
public:
    float m[3][3]; // 변수
    mat3() { // 생성자 함수, 0으로 초기화됨
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                m[i][j] = 0;
            }
        }
    }
};
```

- 행렬의 덧셈 (두 행렬의 크기가 같음)

$$\begin{pmatrix} \boxed{2} & 8 & 3 \\ 2 & 6 & 1 \\ 4 & 1 & 7 \end{pmatrix} + \begin{pmatrix} \boxed{3} & 4 & 1 \\ 2 & 7 & 4 \\ 5 & 6 & 1 \end{pmatrix} = \begin{pmatrix} \boxed{5} & 12 & 4 \\ 4 & 13 & 5 \\ 9 & 7 & 8 \end{pmatrix}$$

- 행렬의 뺄셈

$$\begin{pmatrix} 3 & 8 & 5 \\ 6 & 7 & 4 \\ 5 & 6 & 1 \end{pmatrix} - \begin{pmatrix} 2 & 4 & 3 \\ 2 & 6 & 1 \\ 4 & 1 & 7 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 2 \\ 4 & 1 & 3 \\ 1 & 5 & -6 \end{pmatrix}$$

행렬의 곱셈

- 행렬의 곱셈 (+ - 라는 좀 다름)

$$\begin{pmatrix} 2 & 8 & 3 \\ 2 & 6 & 1 \\ 4 & 1 & 7 \end{pmatrix} \times \begin{pmatrix} 3 & 4 & 1 \\ 2 & 2 & 4 \\ 5 & 6 & 1 \end{pmatrix} = \begin{pmatrix} 37 & 42 & 37 \\ 21 & 26 & 27 \\ 49 & 52 & 15 \end{pmatrix}$$

- (2*3)+(8*2)+(3*5)=37

$$\begin{pmatrix} 3 & 0 & 1 \\ 2 & 5 & 3 \\ 1 & 2 & 4 \end{pmatrix} \times \begin{pmatrix} 0 & 4 & 1 \\ 1 & 3 & 4 \\ 5 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 5 & 13 & 3 \\ 20 & 26 & 22 \\ 22 & 14 & 9 \end{pmatrix}$$

- 행렬의 곱셈

$$\begin{pmatrix} 1 & 3 & 0 \\ 2 & 1 & 4 \end{pmatrix} \times \begin{pmatrix} 2 & 0 \\ 1 & 3 \\ 5 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 9 \\ 25 & 11 \end{pmatrix} \quad 2 \times \boxed{\cancel{3},\ \cancel{3}} \times 2 = 2 \times 2$$

$$\begin{pmatrix} 2 & 0 \\ 1 & 3 \\ 5 & 2 \end{pmatrix} \times \begin{pmatrix} 1 & 3 & 0 \\ 2 & 1 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 6 & 0 \\ 7 & 6 & 12 \\ 9 & 17 & 8 \end{pmatrix} \quad 3 \times \boxed{\cancel{2},\ \cancel{2}} \times 3 = 3 \times 3$$

$$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} = ? \quad (8) \quad 1 \times 3,\ 3 \times 1 = 1 \times 1$$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \times \begin{pmatrix} 3 & 4 \end{pmatrix} = ? \quad \begin{pmatrix} 3 & 4 \\ 6 & 8 \end{pmatrix} \quad 2 \times 1,\ 1 \times 2 = 2 \times 2$$

```
mat3 operator*(mat3 a) {
    mat3 result; // 기본 생성자에서 0으로 초기화됨
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 3; k++) {
                result.m[i][j] += m[i][k] * a.m[k][j];
            }
        }
    }
    return result;
}
```

코드는 시험범위

덧셈은

```
for(int i ~~~){
    for(int j ~~~){
        result.m[i][j] = m[i][j] + a.m[i][j];
    }
}
```

단위행렬

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 4 & 1 \end{pmatrix}$$

어떤 행렬이든 저렇게 생긴걸 곱하면
원본이 나옴

단위 행렬의 생성

```
void LoadIdentity () {
  for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
              if(i==j)            m[i][j]=1;
              else                m[i][j]=0;
        }
  }
}
```

전치행렬

- 행렬 M이 주어졌을 때 행과 열을 바꿔 놓은 것
- M^T 로 표기

회전축

- $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$

- $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}^T = ?$ $\begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$

전치행렬

```
mat3 Transpose() {
    mat3 result;
    for (int i = 0; i < 3; i++) {
        for(int j=0; j<3; j++){
            result.m[i][j]= m[j][i]; // 행,열 위치가 바뀜
        }
    }
    return result;
}
```

역행렬

$3 \times \frac{1}{3} = 1$

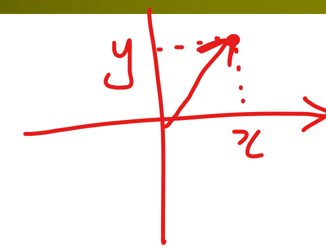
- A × B= I 라면, B는 A의 역행렬
- $B = A^{-1}$ 로 표기
- $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \underline{\left[\frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}\right]} = I$
- $\begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix}^{-1} = \frac{1}{2} \begin{pmatrix} 2 & -1 \\ -4 & 3 \end{pmatrix} \neq \begin{pmatrix} 3 & 1 \\ 4 & 2 \end{pmatrix}^{T}$

벡터

- 2차원 벡터는 (x,y) 또는 $\begin{pmatrix} x \\ y \end{pmatrix}$ 로 표기
- 3차원 벡터는 (x,y,z) 또는 $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ 로 표기
- 벡터는 행렬의 일종으로 볼 수도 있음

```
class vec3 {
public:
    float m[3];
    vec3() {
        for (int i = 0; i < 3; i++) {
            m[i] = 0;
        }
    }
};
```

벡터의 크기



- 2차원 벡터 길이 $\|(x_1\ , y_1)\| = \sqrt{{x_1}^2 + \ {y_1}^2}$
- $\|(3\ , 4\ )\| = 5$
- 3차원 벡터 길이 $\|(x_1\ , y_1, z_1)\| = \sqrt{{x_1}^2 + \ {y_1}^2 + \ {z_1}^2}$
- $\|(3, 4, 5)\| = ?$

$\sqrt{9+16+25} = \sqrt{50}$

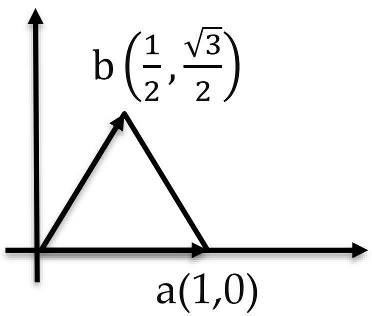
- $(x_1\ , y_1) \cdot (x_2\ , y_2)\ =\ x_1 x_2 + y_1 y_2$
- $(1, 0) \cdot \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\ =\ \frac{1}{2}$
- $(x_1\ , y_1, z_1) \cdot (x_2\ , y_2, z_2)\ =\ x_1 x_2 + y_1 y_2 + z_1 z_2$
- $(1, 2, 3) \cdot (-1\ , 0, 2\ ) =$? $-1 + 0 + 6 = 5$

```
float vec3::DotProduct(vec3 b) {
    return m[0] * b.m[0] + m[1] * b.m[1] + m[2] * b.m[2];
}

float vec3::operator*(vec3 b) {
    return m[0] * b.m[0] + m[1] * b.m[1] + m[2] * b.m[2];
}
```

내적

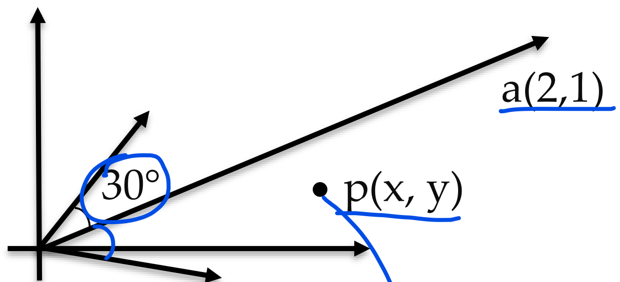
- $\boxed{a \cdot b = \|a\| \|b\| \cos\theta}$ 오른쪽을 왼쪽으로 간단히 풀 수 있다



- $(1,0) \cdot \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right) = \|(1,0)\| \times \left\|\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)\right\| \times \cos 60$
- $\frac{1}{2} = \frac{1}{2}$

내적 예제

- 점 p(x,y)가 주어졌을 때, 범위에 들어오는지 리턴하는 함수를 작성하라



$$a \cdot p = \|a\|\|p\| \cos\theta$$

$$2x + y = \sqrt{5} \times \sqrt{x^2+y^2} \times \cos 30^\circ$$

```
bool IsInside(float x, float y) {
```

$$\text{if} \left(2x + y <= \sqrt{5} \times \sqrt{x^2+y^2} \times \cos 30^\circ \right) \text{ return } 1;$$

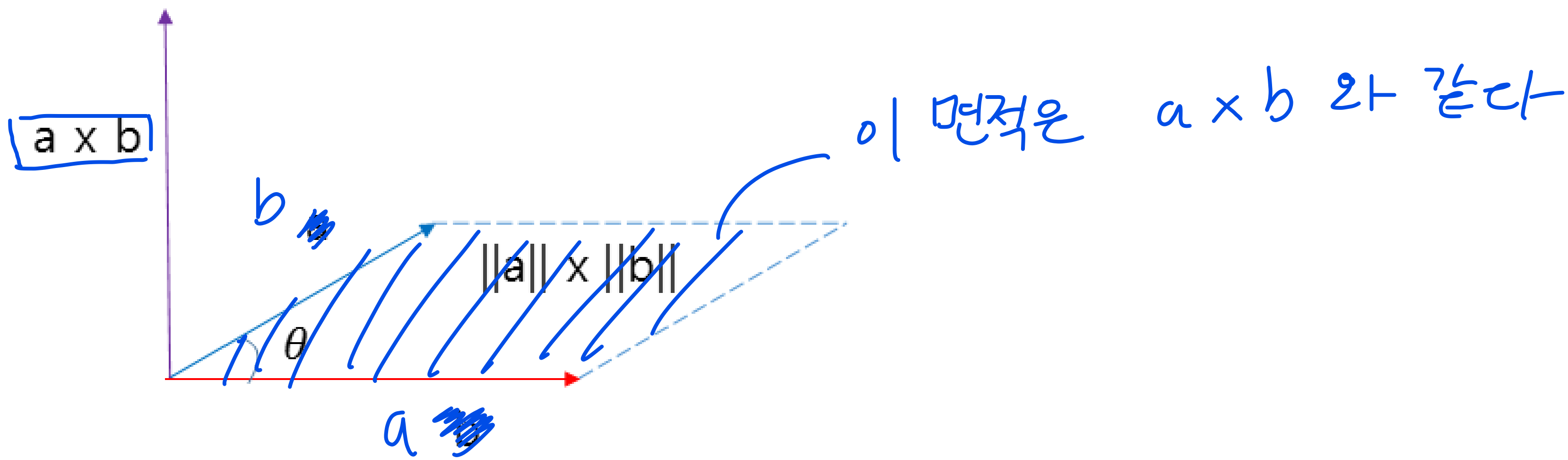
각이 작아져도 맞긴하니까 <=

$$\cos 60^\circ < \cos 30^\circ = \frac{\sqrt{3}}{2} < \cos 29^\circ \;\ldots\ldots$$

```
}
```

외적

- 두 개의 3차원 벡터 a와 b의 벡터곱은 a X b 로 표기
- a와 b에 모두 수직인 또 다른 3차원 벡터



외적

- $A = (\, a, b, c), \quad B = (d, e, f)$
- $A \times B = (\, b * f \; - c * e, \; c * d \; - a * f, \; a * e \; - b * d)$

- $x \times y = z \,, \qquad y \times z = x, \qquad z \times x = y$
- $y \times z = -z \,, \qquad z \times y = -x, \qquad x \times z = -y$
- $x \times x = 0 \,, \qquad y \times y = 0, \qquad z \times z = 0$

- $(2x + y) \times (y + z)$
- $= 2z \; - 2y + 0 + x = x \; - 2y + 2z$

- $(2, 1, 0\,)$
- $\times\, (0, 1, 1\,)$
- —————
- $(1, -2, \; 2)$

$$\begin{array}{cccc} & a & b & c \\ \times & d & e & f \\ \hline & bf-ce & cd-af & \boxed{ae-bd} \end{array}$$

구하고자 하는 위치 빼고 크로스, 가운데는 예외

외적

```
vec3 vec3::CrossProduct(vec3 b) {
    vec3 result;
    result.m[0] = m[1] * b.m[2] - m[2] * b.m[1];
    result.m[1] = m[2] * b.m[0] - m[0] * b.m[2];
    result.m[2] = m[0] * b.m[1] - m[1] * b.m[0];
    return result;
}
```