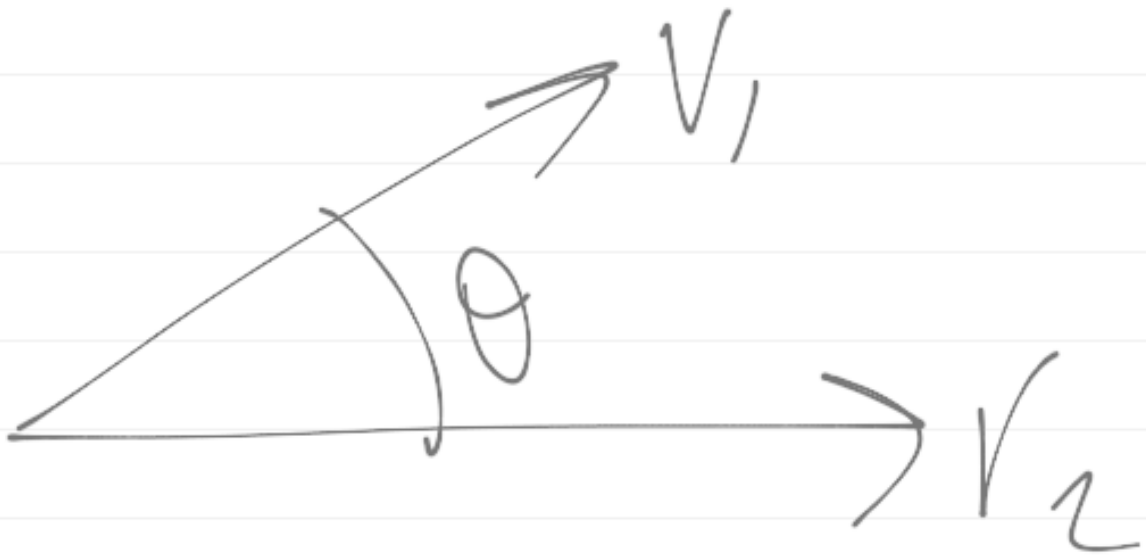
```
Vec_sum(Vec_t v1,
        Vec_t v2,
        Vec_t result)
{
    int i = 0;
for(i=0; i<3; i++)
  result[i] = v1[i] + v2[i];
}
```



$$\frac{V_1 \cdot V_2}{\|V_1\| \|V_2\|} = \cos\theta$$

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} =$$

$$x_1 * x_2 + y_1 * y_2 + z_1 * z_2$$

```
double
Vec_dot(...v1, ...v2)
{
    int i;
    double sum = 0.0;

    for(i=0; i<3; i++)
        sum += v1[i] * v2[i];

    return sum;
}
```

$$||v|| = \left|\left|\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right|\right| = \sqrt{x^2 + y^2 + z^2}$$

$$= \sqrt{x*x + y*y + z*z}$$

```
Vec_len(v) :
    return (sqrt(Vec_dot(v, v)))
```

```
#include <math.h>
```

vec_scale(double s, v1, res)

a scalar (just a number)

for(...)

res[i] = s * v1[i]

main.c:

Vec_unit(v1, v1) } Vec_unit usage

//normalize v1

↑

means to divide every element of v1 by its length

Vec_unit(v1, v2) } in vects.c

Vec_scale(1.0/vec_len(v1), v1, v2);

Usage: (in main.c)

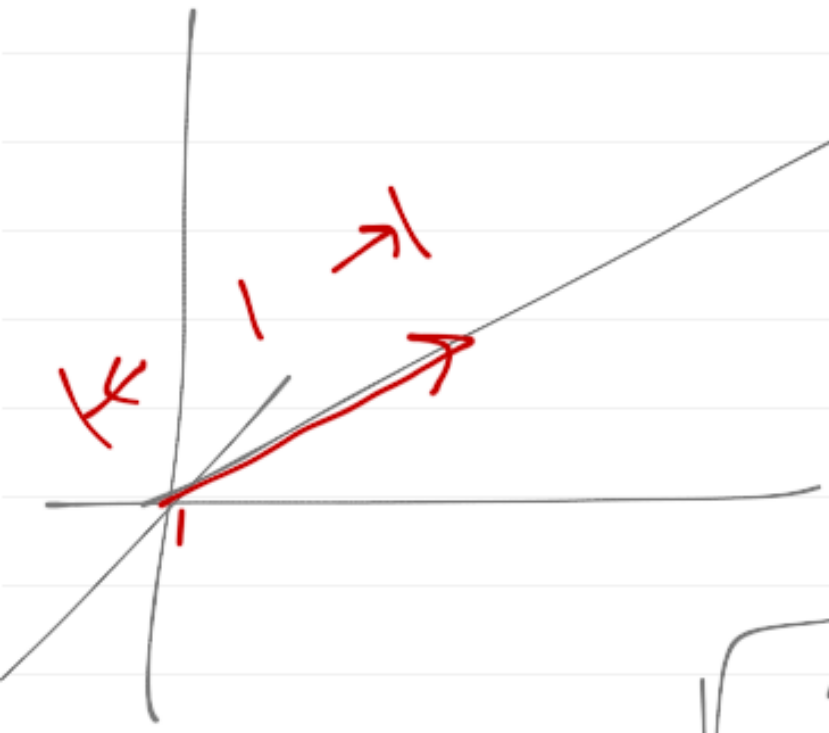
```
vec_print(stdout,
    "V1 scaled by its
     length:", v3);
```

---

implementation (in vector.c):

```
vec_print(FILE *fd,
          char *s,
          vec_t v)
```

```
fprintf(fd, "%s ", s);
for (i=0; i<3; i++)
    fprintf(fd, "%8.3lf",
            v[i]);
fprintf(fd, "\n");
```



$v_1 = (3, 4, 5)$

$\|v_1\| =$

$\sqrt{3^2 + 4^2 + 5^2}$

$= 7.071$

$$\frac{v_1}{\|v_1\|} = \left(\frac{3}{7.071}, \frac{4}{7.071}, \frac{5}{7.071}\right)$$

$= (.424, .566, .707)$

$$\frac{\vec{a} \cdot \vec{b} =}{\|\vec{a}\| \, \|\vec{b}\|} = \cos\theta$$

- to get θ, angle between 2 vector $\vec{a}, \vec{b}$,

  ① normalize a, b

  ② get dot prd.

  ③ take $\cos^{-1}$

$$\theta = \cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}\right)$$

theta =

vel_unit(v1, v1n)

vel_unit(v2, v2n)

acos(vel_dot(v1n, v2n))