

14/9

macro-functi  
# define N 100  
# define min(a,b) (a < b ? a : b)  

---

static int ~~100~~ N = 100;  
const int N = 100;  
char ~~int~~ a[N];  $\equiv$  char a[100];  

K = K \* N;

  
K = K \* N;

$$\cos \theta = \frac{A \cdot B}{|A| \cdot |B|}$$

struct ✓  
typedef struct vector {  
 float x;  
 float y;  
 float z;  
} vector;

main().

```
{ vector a, b;  
  float cosF;
```

```
  cosF =  
    scanf("%f%f%f", &a.x, &a.y, &a.z);  
    scanf("%f%f%f", &b.x, &b.y, &b.z);
```

$$\cos F = \frac{a.x * b.x + a.y * b.y + a.z * b.z}{\sqrt{(a.x^2 + a.y^2 + a.z^2)} * \sqrt{(b.x^2 + b.y^2 + b.z^2)}}$$

```
    sqrt(a.x * a.x + a.y * a.y + a.z * a.z)  
    sqrt(b.x * b.x + b.y * b.y + b.z * b.z)
```

```
    printf("%f", cosF);
```

char a[100], b[100];

→ TASK. is to find the smaller one  
in lexicographic order.

→ assumption - all are small cases.

(assumption  
assume.)

int ~~void~~ smaller(char a[100], char b[100])  
 // put the smaller in a and the larger in b, return  
 // both are strings of size n  
 char temp;

for (i=0; i < n; i++)  
 { If (a[i] != b[i]).  
 break;

}  
 { If (b[i] < a[i]).  
 for (j=i; j < n; j++)  
 { temp = a[j];  
 a[j] = b[j];  
 b[j] = temp;  
 }

return (1)  
 (2)

sketch node. f.

float { int a;  
 int b;  
 char c;  
 }

point in a.  
 cultural space

14/9

```
typedef struct node {
    float x;
    float y;
} node;
```

```
main ( )
{ node a[100];
```

/\* sort on x. if there is tie sort on y \*/

→ (1,2) (1,7) (1,5) (2,6) ~~(1,6)~~ ~~(3,5)~~

→ (1,2) (1,5) (1,6) (1,7) (2,6) (3,5).

→ (1,2) (1,7) (1,5) (1,6) (2,6) (3,5)

Sort-ascending (node a[], int n).

```
{
for (j=0; j<n; j++)
{
min = a[j].x, minp = j;
```

```
for (i=j; i<n; i++)
```

```
{ if (a[i].x < min)
```

```
{ minp = i; min = a[i].x;
```

```
{ swap
```

```
{ a[j].x = a[i].x;
a[i].x = min; } swap;
}
```

↓



Sort - secondary (node a[], int n).

{ a = 0;

for (~~a=0~~; i < n; i++).

~~if (a[i].x != a[i-1].x) || (b == n)~~

~~for (i = 0; i < n; i++).~~

{ min = a[i].y; minp = i;

for (j = i; j < n; j++).

{ if (a[j].y < min).

{ minp = j; min = a[j].y;

~~a[minp].x = a[i].x;~~

~~a[minp].y = a[i].y;~~

~~a[minp].x = a[i].x;~~

~~a[i].y = min;~~

~~a[i].x = minp;~~

~~a = b;~~

Selection sort