

Inner Loop Shifts “Things” to Find Place for **current**

```
for (index = sorted - 1; 0 < index;
    index--) {
```

Loop structure is again identical to the integer version of the sort.

```
    // inner loop body performs
    // comparison and may copy
    // one thing to adjacent place
```

```
}
```

21

Comparison Requires **is_smaller**

```
if ((*is_smaller) (current,
    array + (index - 1) * size)) {
    memcpy (array + index * size,
            array + (index - 1) *
                size, size);
```

```
} else {
    break;
}
```

Use **is_smaller** to compare **current** with the thing in array element **index - 1**.

22

Keep Shifting Things Until **current**'s Place is Found

```
if ((*is_smaller) (current,
    array + (index - 1) * size)) {
    memcpy (array + index * size,
            array + (index - 1) *
                size, size);
```

```
} else {
    break;
}
```

If **current** is smaller, copy array element **index - 1** over array element **index**.

23

When **current** is Not Smaller, the Loop is Done

```
if ((*is_smaller) (current,
    array + (index - 1) * size)) {
    memcpy (array + index * size,
            array + (index - 1) *
                size, size);
```

```
} else {
    break;
}
```

Otherwise, we found the right place!

24