Return Value from `isort` Indicates Success or Failure

Returns 1 on success, or 0 on failure.

```
int32_t isort
    (void* base,
     int32_t n_elts,
     size_t size,
     int32_t (*is_smaller)
         (void* t1, void* t2));
```


Function Pointer: a Function Signature, Plus an Asterisk

Focus now on the function pointer argument:

Function pointer's identifier preceded by * and surrounded by parentheses.

```
int32_t (*is_smaller)
    (void* t1, void* t2)
```

return type

argument list

Function Pointer Meaning Must Be Documented

Meaning of the function pointer argument
- **must be documented**,
- just like other parameters!

`is_smaller`
- compares two array elements, `t1` and `t2`
- and returns
  - non-zero (true) if `t1 < t2`,
  - 0 (false) if `t1 ≥ t2`.

Change Variables to Support "Things" of Any `size`

Now for code (based on the integer version):

array pointer for pointer arithmetic (`base` has type `void*`)

```
char*   array = base;
void*   current;
int32_t sorted;
int32_t index;
```

space for one "thing" (size varies, so space is allocated dynamically)

same loop indices as used for sorting integers