Read and Build a Pyramid Tree from a File

Now, let's **reconstruct a pyramid tree from a binary file.**

```
pyr_tree_t* read_pyr_tree_binary
    (const char* fname)
{
    FILE*       in;
    pyr_tree_t* p;
    int32_t     count;
```

Returns new tree or NULL.

file name

input stream

new pyramid tree

number of nodes in file

Open File for Reading, Then Read Number of Nodes

```
if (NULL == (in =
             fopen (fname, "r")) ||
    1 != fread (&count,
                sizeof (count), 1, in)) {
    if (NULL != in) {
        fclose (in);
    }
    return 0;
}
```

Open file for reading.

If file open succeeds, read number of nodes in file.

On Failure, Try to Close Stream, Then Return Failure

```
if (NULL == (in =
             fopen (fname, "r")) ||
    1 != fread (&count,
                sizeof (count), 1, in)) {
    if (NULL != in) {
        fclose (in);
    }
    return 0;
}
```

If either fails, try to close stream, then return failure.

Allocate Space for Pyramid Tree and Node Array

Allocate space for pyramid tree.

```
if (NULL ==
    (p = malloc (sizeof (*p))) ||
    NULL == (p->node = malloc
    (count * sizeof (p->node[0])))) {
    if (NULL != p) { free (p); }
    fclose (in);
    return NULL;
}
```

Allocate space for node array.

