On Failure, Free Tree and Close Stream, Then Return

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

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

Read Node Array from Input Stream

Write number of nodes into pyramid tree.

```
p->n_nodes = count;
if (p->n_nodes != fread
      (p->node, sizeof (p->node[0]),
       p->n_nodes, in)) {
     free_pyramid_tree (p);
     fclose (in);
     return NULL;
}
```

Read node array from stream.

If node array read fails, free tree, close stream, and return failure.

Clean Up and Return the New Pyramid Tree

Discard the return value (explicit).

```
(void) fclose (in);
return p;
```

Close the input stream

Return the new pyramid tree.

Time for Another Think-Pair-Share

As before, let's do a group exercise in lecture.

The process:

1. I give you a problem.
2. You form groups of 3-4 people.
3. Talk about ways to solve the problem.
4. Once enough of the groups have finished, one group volunteers to share their answer.
5. We go over the group's answer together.

