

## Array Elements

Since Pascal counts from '1' and C counts from '0' you'll have to manually adjust array element designations.

In Pascal:

```
procedure Normalize (var ThePoint: Coordinates);
var
    Length: Real;
begin
    Length := SQRT(ThePoint[1] * ThePoint[1] +
                  ThePoint[2] * ThePoint[2] +
                  ThePoint[3] * ThePoint[3]);

    ThePoint[1] := ThePoint[1] / Length;
    ThePoint[2] := ThePoint[2] / Length;
    ThePoint[3] := ThePoint[3] / Length;
end;
```

shows the array "Bvert" to have 3 elements. It's still 3 in C but are designated 0, 1 and 2 instead of 1, 2 and 3. So the above would become:

```
void Normalize (Coordinates *ThePoint);

double Length;

Length = sqrt(ThePoint[0] * ThePoint[0] +
              ThePoint[1] * ThePoint[1] +
              ThePoint[2] * ThePoint[2]);
ThePoint[0] = ThePoint[0] / Length;
ThePoint[1] = ThePoint[1] / Length;
ThePoint[2] = ThePoint[2] / Length;
```

All array element designations reduced by one. There ARE some that don't get changed. You'll have to play it by ear to see which is which. The key is in the declaration. If it says "typedef short widget [3]" then you know that "widget[1]" is one too high.

While we're on arrays, CTools™ doesn't bracket two dimensional ones right. If the Pascal says "widgets [1, 20]" then will, too. So you have to replace the ",", with "]" between them, to make "widgets[1][20]". The compiler will find these for you. If you have a lot of them, you may be able to Find all "[1, " and replace with "[1]" to do it faster.