Meagin Arrocha

1. int \*p1 = &x;
2. \*p1 = 33;
3. int \*p2 = &arr[0];
4. \*p2 = 5; or p2[0] = 5;
5. p2[2] = 15;
6. \*(p2 + 3) = 35;
7. p = p + 2;
8. \*p2 = 77;
9. If(\*p == \*q)
10. If( p == q)
11. p1 = new int;
12. delete p1;
13. p1 = new int [x];
14. delete [ ] p1;
15. It has an orphan because p made a new dynamically allocated integer then pointed at it, then made another DAI and pointed at it before deleting the first one making for bad programming. Line 2 is trying to change the value of what p is pointing to when p isn’t pointing to anything.

Line 8 is trying to place a value in something it’s not pointing at because it deleted the second DIA and the first one is an orphan so it’s a dangling pointer.

1. Dynamically the memory isn’t known until runtime but static is the opposite because you know it before runtime. It has both because it is better if you already know how much memory to set aside before runtime, but you don’t always know so you can still be okay in either option.