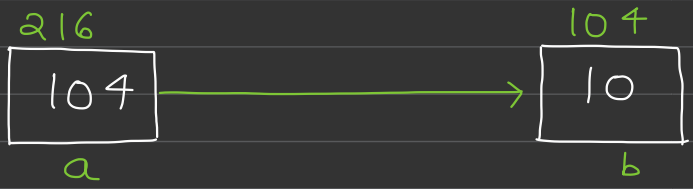
Quiz-6 Detailed Solutions

1. `int a = 5;`
   `int *ptr = &a;`
   pointer declaration & initialization

2. ampersand (&) operator is used to get the address of variable.

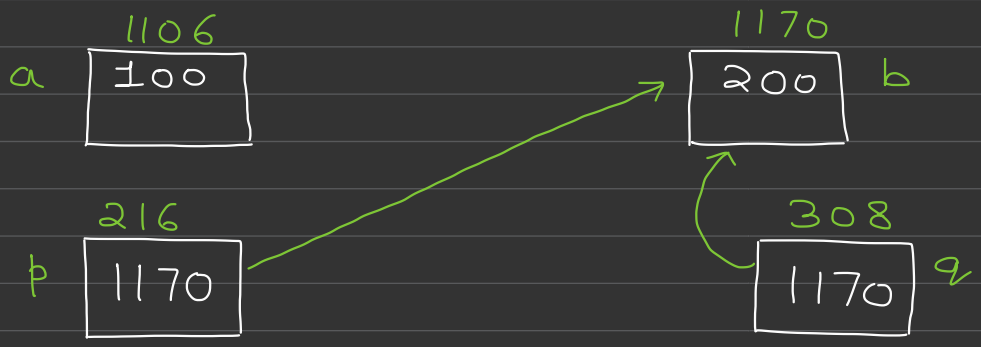
3. `int b = 10;`
   `int *a = &b;`



a → address of b.

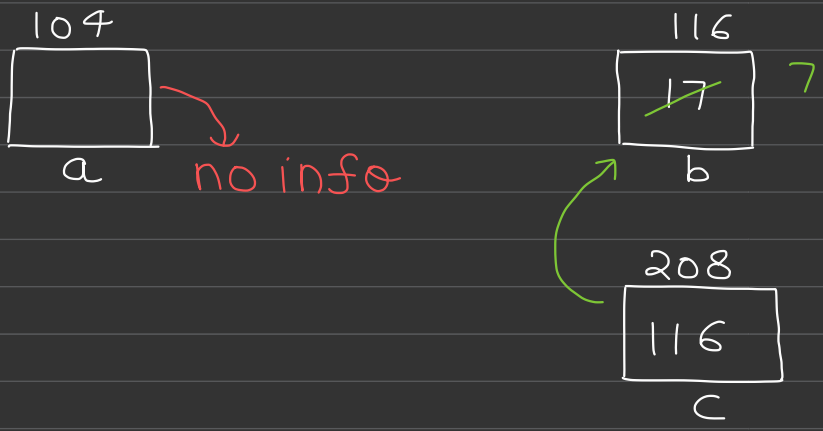
4. `int a = 100, b = 200;`
   `int *p = &a, *q = &b;`
   `p = q;`

After p = q



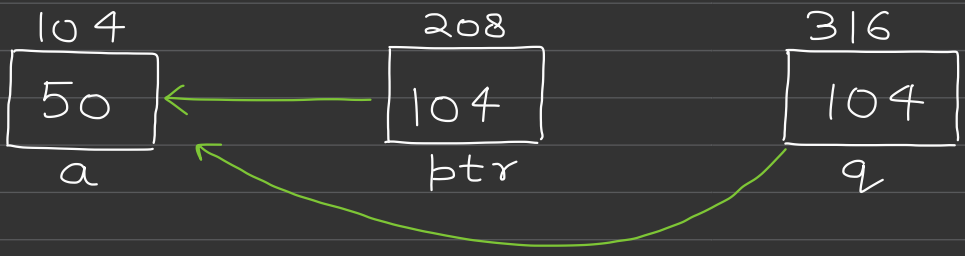
p now points to b.

5.



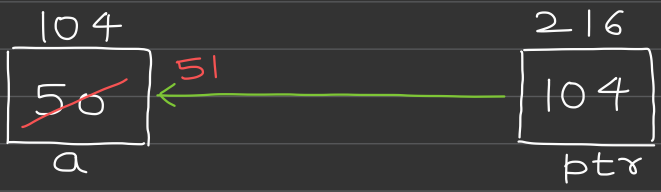
*c = 7 will modify b to 7 from 17.

6.



$(*q)++ \Rightarrow 50+1=51$ → a

7.



cout << (*ptr)++ << " "; → 50

cout << a << endl; → 51

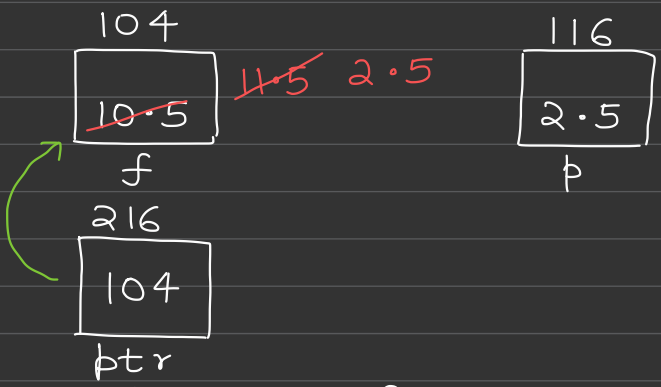
8.

int *ptr = 0; X

↳ null_ptr

int *ptr = null_ptr;

9.



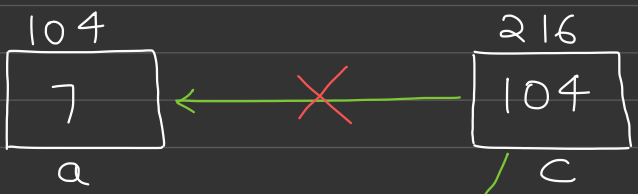
(*ptr)++ ⇒ f = 10.5 + 1 = 11.5

*ptr = p;

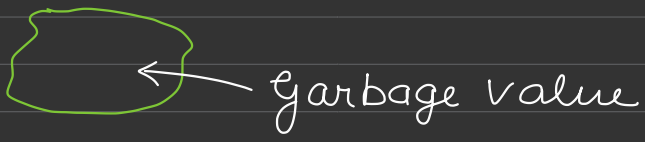
*ptr → 2.5 p → 2.5

f → 2.5

10.



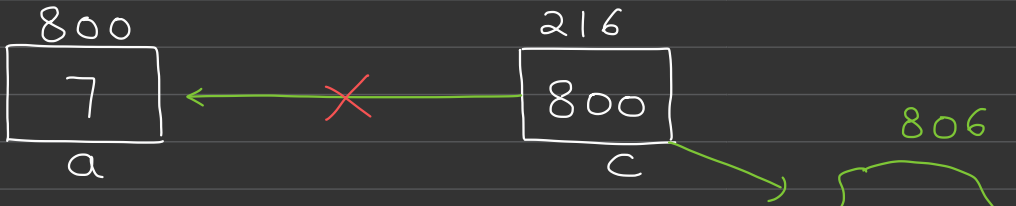
c = c+1



a → 7

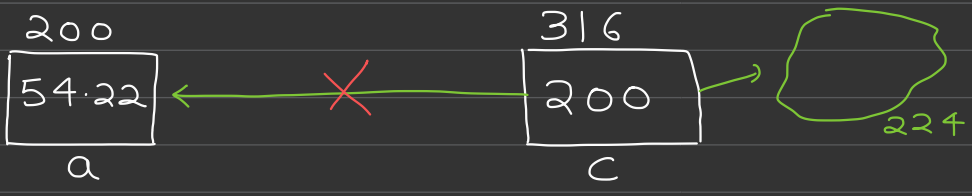
*c → garbage value

11.



c → 800 + 2×3 = 806

12.



c = 200 + (8×3) = 200 + 24 = 224

13. int a[5];
sizeof(a); → 5×4 = 20 bytes
int *c;
sizeof(c); → 8 bytes (pointer size)

14. `int a[] = {1, 2, 3, 4}`

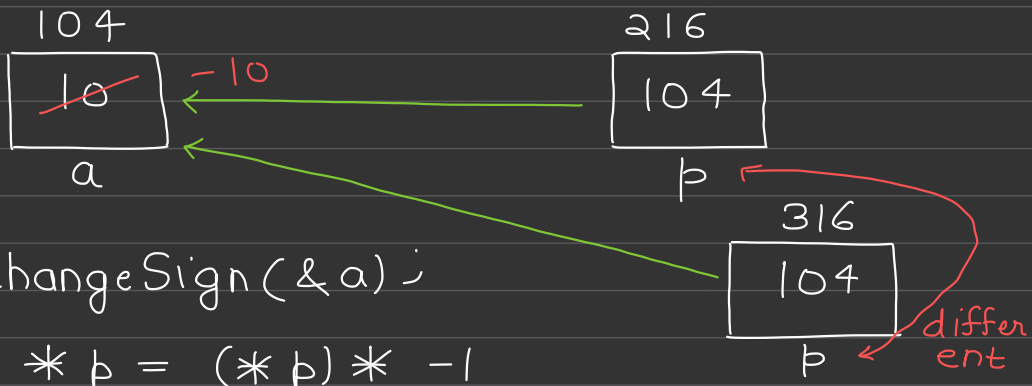
a → 1, a+3 → 4

`*a = 1`

`*(a+3) = 4`

15. Error as a is a constant pointer.

16.



`ChangeSign(&a);`

`*p = (*p) * -1`

17. Same concept as that of above. Value is changed to 11.

18. Time complexity = $O(n \log \log n)$ of Sieve of Eratosthenes.

19. We have to cross out multiples of the primes as they won't be prime.

20. Size of array is n+1 i.e $O(n)$ is the space complexity.