1. True
   1. head is set equal to malloc(sizeof(Node))
2. False
   1. it returns counter, which is an int. An int could be stored in a node in the linked list, but, there is no guarantee that counter will equal age
3. True
   1. This will allow head to increment through the linked list
4. False
   1. Age is an int, while malloc(sizeof(Node)) would return a pointer. A pointer and int aren’t equal
5. True (False?)
   1. The first value temp is set to is current, which the first value current has is h
6. True
   1. Free(temp) is in the while loop, and is freeing a pointer, which is 8 bytes.
7. True
   1. Second->next contains the address of third, which links the two
8. True
   1. Both head and third are Node pointers
9. True
   1. We allocated ram for head, second, and third, but never freed the memory, so there will be a memory leak
10. True
    1. h is never used outside of the context of current, so instead we could just replace current with h
11. False
    1. If we free current, that will free the node that is going to be used next, causing current to equal NULL, and the list to stop running through. We need to free temp as that frees the node that was just used.