LINKED LISTS

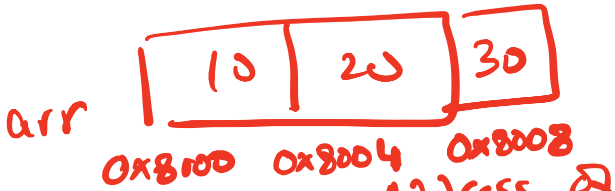
Problem Solving with Computers-I

```
#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook\n";
    return 0;

}
```

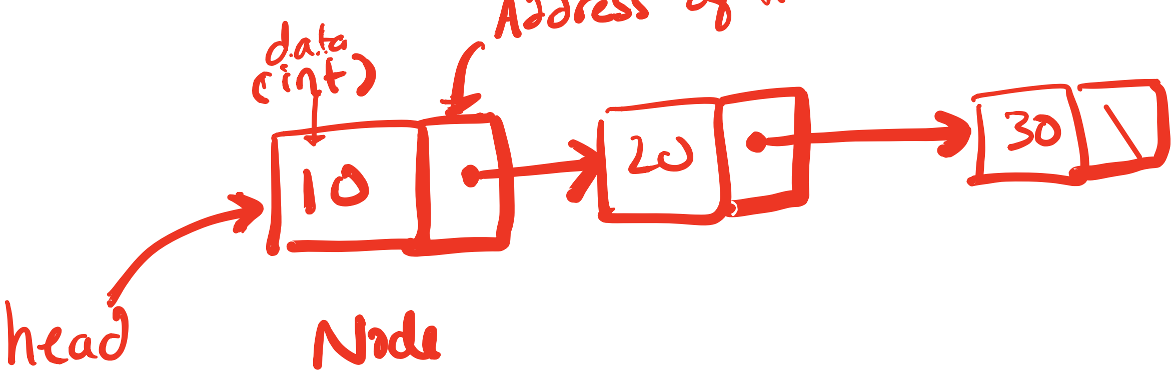
Store a sequence of numbers or names or structs or any data

1. Array : Fixed Size
   int arr [100]; // Only store a 100 numbers

2. Linked list : Variable size
   Main difference compared to arrays
   Elements are not next to each other in memory



"Node"
↕
"boogie"



Representing a Node

```
struct Node {
    int data;
    Node* next;

};
```

Create a singe node linked list

Node n1;

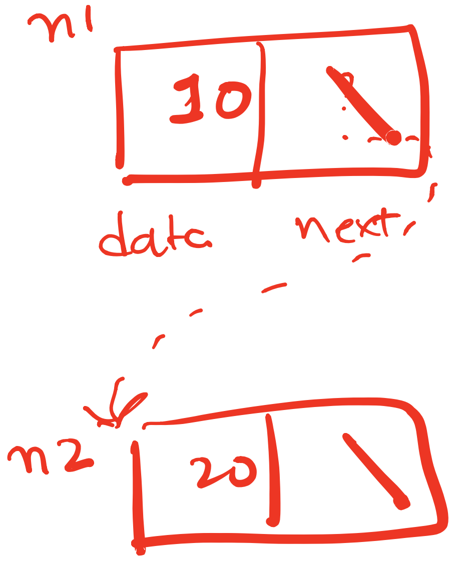
n1. data = 10;

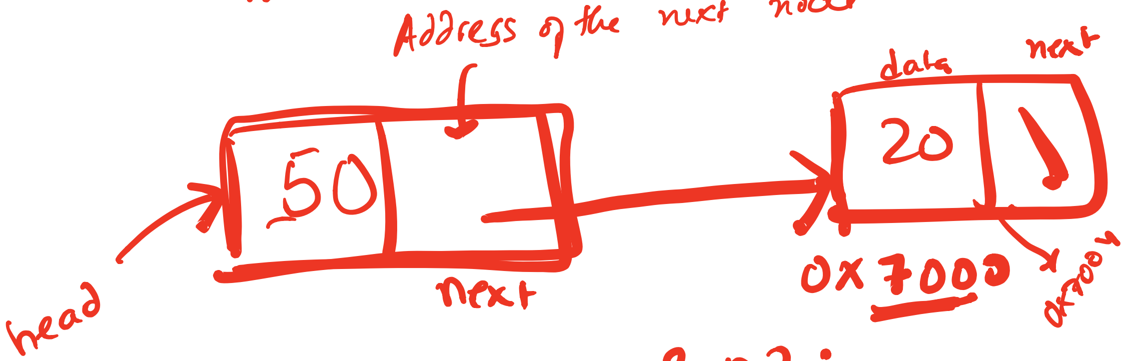
n1. next = 0;

Node n2;

n2. data = 20

n2. next = 0;





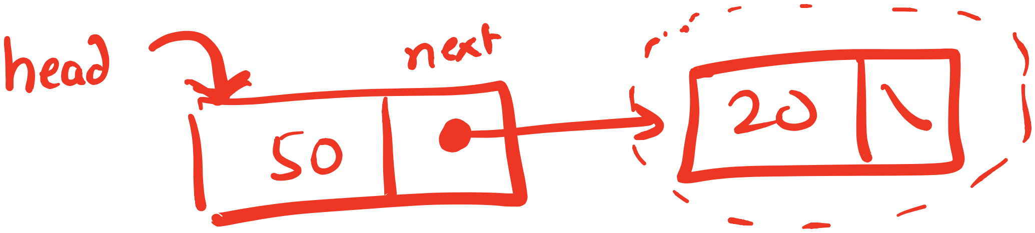
n1. next = &n2;

n1. data = 50;

Node *head = &n1;

Print the data of both nodes using the variable head

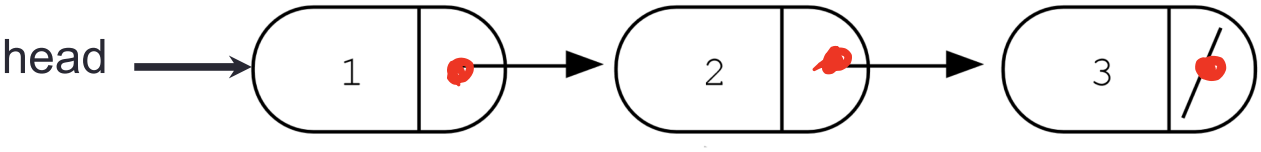
cout<<(*head). data ;



```
cout << head -> data;

cout << head -> next -> data;

cout<< head -> next -> next;

cout<<head -> next -> next -> data
;
```

Accessing elements of a linked list

```
struct Node {
     int data;
     Node *next;
};
```



Assume the linked list has already been created, what do the following expressions evaluate to?

1. head->data
2. head->next->data
3. head->next->next->data
4. head->next->next->next->data

is null

Dereferencing a null pointer

(Seg fault!)

A. 1
B. 2
C. 3
D. NULL
[✓ E. Run time error]

Creating a small list

- Define an empty list
- Add a node to the list with data = 10

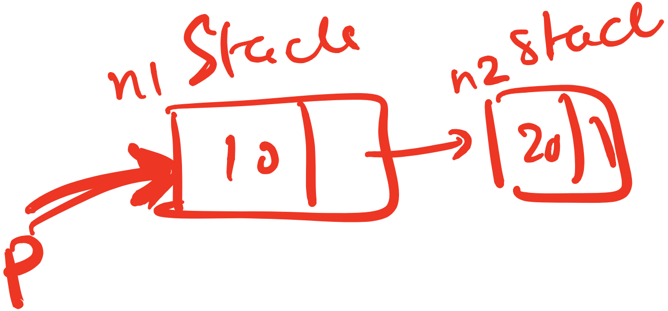
```
struct Node {
        int data;
        Node* next;
};
```

Heap vs. stack

```
Node* createSmallLinkedList(int x, int y){
    Node* head = NULL;
    Node n1 ={x, NULL};
    Node n2 ={y, NULL};
    head = &n1;
    n1->next = &n2;
    return head;
}
```

node *p;

p = creat... (10, 20)

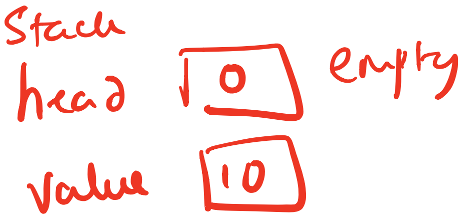


Does the above function correctly return a two-node linked list?

A. Yes

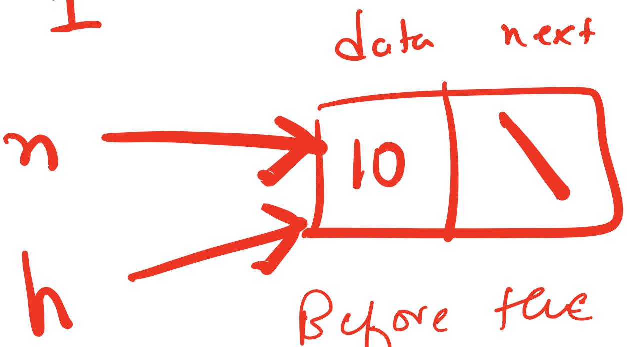
[✓ B. No]

The nodes are removed from the stack after the function returns



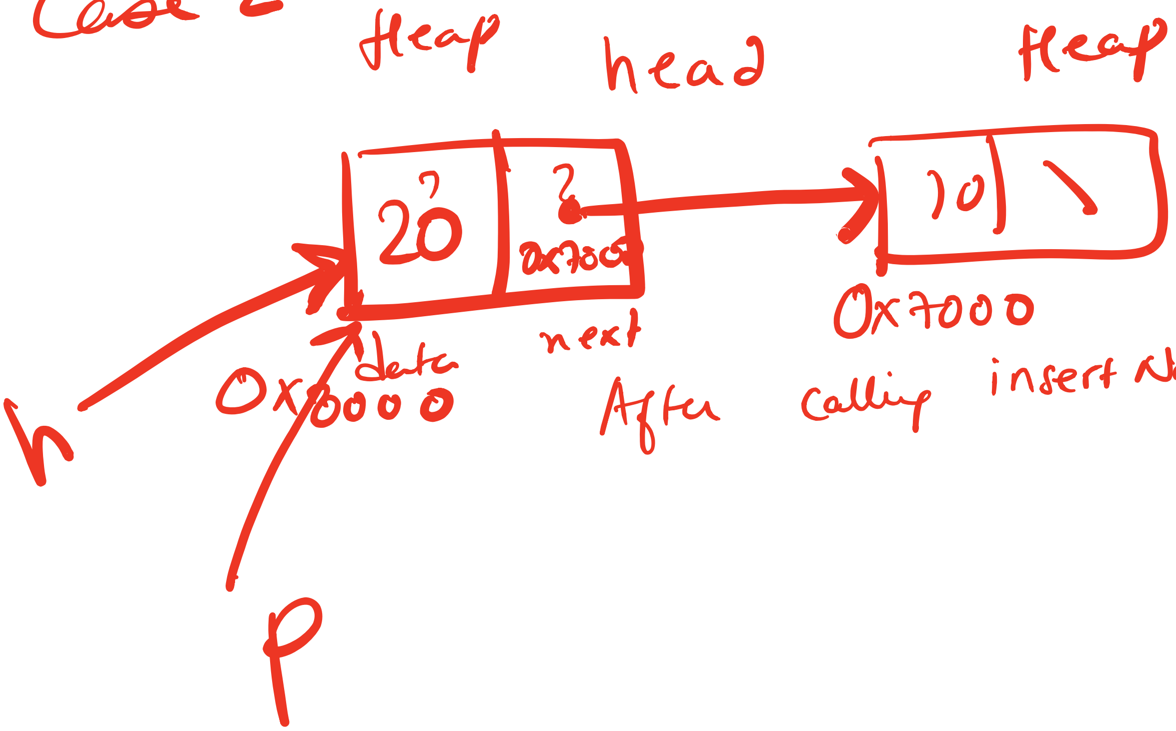
Heap

Case 1



Before the call to insert

Case 2



After calling insert Node

Creating a small list

- Define an empty list
- Add a node to the list with data = 10

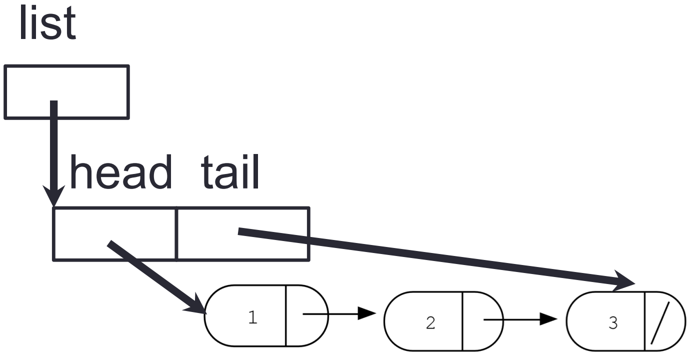
```
struct Node {
     int data;
     Node* next;
};

struct LinkedList {
     Node* head;
     Node* tail;
};
```

Inserting a node in a linked list

```
void insert(LinkedList* h, int value) ;
```

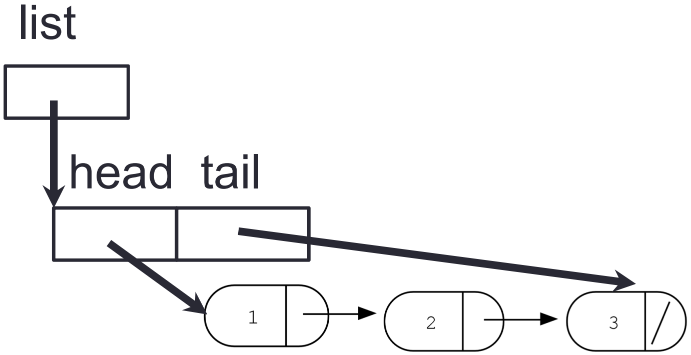
Iterating through the list

```
int count(LinkedList* list) {
   /* Find the number of elements in the list */









}
```



Deleting the list

```
int freeList(LinkedList * list) {
   /* Free all the memory that was created on the heap*/









}
```



Next time

- Memory-related errors
- Double-linked lists