

LINKED LISTS

Problem Solving with Computers-I

C++

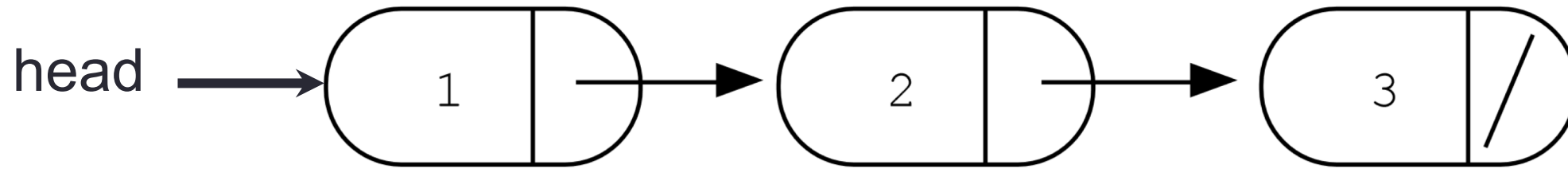
```
#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook!";
    return 0;
}
```

GitHub



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

- 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;  
}
```

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

- A. Yes
- B. No

Creating a small list

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

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

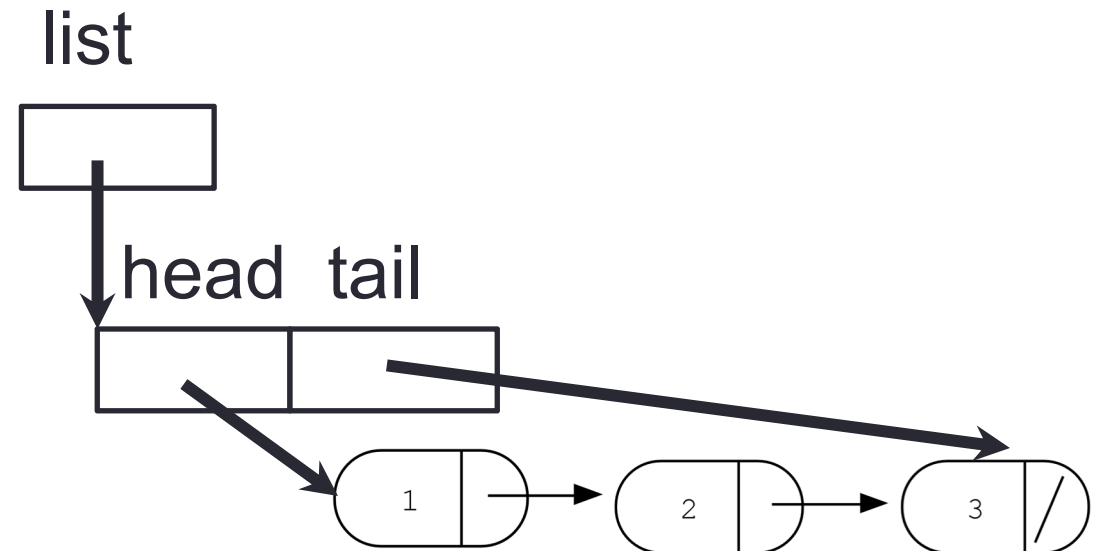
struct LinkedList {
    Node* head;
    Node* tail;
};
```

Inserting a node in a linked list

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

Iterating through the list

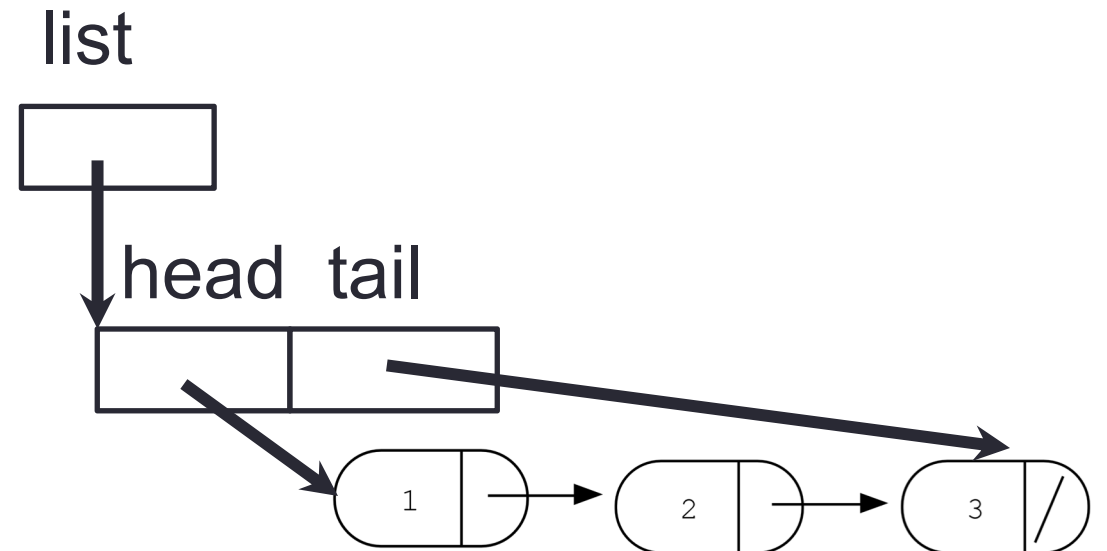
```
int count(LinkedList* list) {  
    /* Find the number of elements in the list */  
}
```



```
}
```

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