

CS162 Final

- x 1 hr 50 min
- x closed book, closed notes
- x Small code fragments

TOPICS

- ✓ • C++ if, loops, functions
- ✓ • classes
 - constructors
 - destructors
 - public / private
 - member functions
- ✓ • Pointers & dynamic memory

✓✓ • Linear Linked Lists

- build ←
- traverse ←
- destroy ←
- remove ←
- add ←

X • Recursion

2a.

```
class list
{
    public:
        ~list();
        list();
        int search ( char [ ] );
        int add ( char [ ], char [ ] );
        int remove ( char [ ] );
        int print ( _____ );


    private:
        node * head;
};
```

```

2 b. list::list()
    { head = NULL;
    }

```

2 c. Destructor - Deallocate all dyn. memory

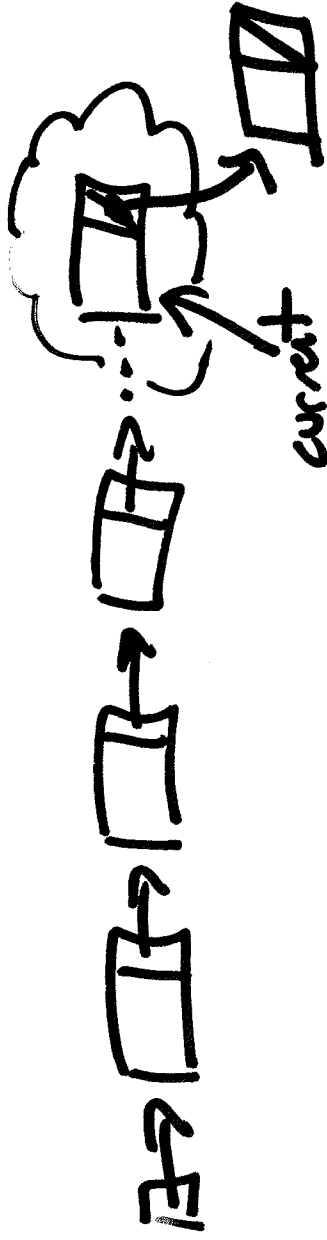


```

list::list()
{
    node *temp = head;
    while (head != NULL)
    {
        temp = head -> next;
        delete head;
        delete head;
        head = temp;
    }
}


```


30/10/17



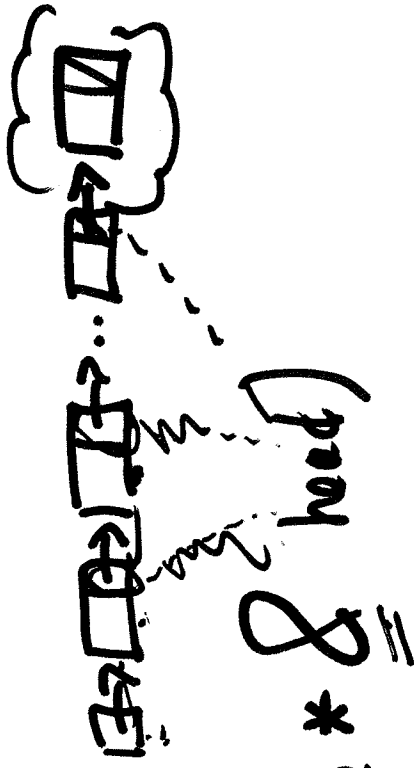
Assuming Empty List

```
node * current = head;
while ( current->next != NULL )
    current = current->next;
// now current is pointing to last node;
current->next = new node;
current = current->next;
current->next = NULL
// save the data
```

30/10/22

Using

Recursion



```
void add-at-end (node * & head)  
{  
    if (head == NULL;  
    {  
        head = new node;  
        // store my data  
        head->next = NULL;  
    }  
    else add-at-end ( head->next );  
}
```

Q17 #3

Using Recursion

(node * head)

```
node * add-at-end (node * head)
{
  if (head == NULL)
  {
    head = new node;
    // save the data
    head->next = NULL;
    return head;
  }
}
```

```
    }
    else
    {
      head->next = add-at-end (head->next);
      // use the returned value
    }
  }
  return head;
}
```


30 First... Display every node in a LLL

30

```
node * current = head;
while (current != NULL)
{
    cout << current->data;
    cout << endl;
    current = current->next;
}

```

Now, display every other

```
while (current != NULL)
{
    cout << current->data << endl;
    current = current->next->next;
}

```

Has a problem!

Seg fault!

```
while (current != null)
{
    cout << current->data << endl;
    current = current->next;
    if (current = current->next)
        current = current->next;
}
}
```

...

Other

- Display the Last item in a LLL
- Display the Second to the last item in a LLL
- Remove the last item in a LLL
- Insert a node at beginning of LLL
- Count the number of nodes in a LLL
- Remove the first node in a LLL