

Applying a recursive solution to the

# TOWERS OF HANOI

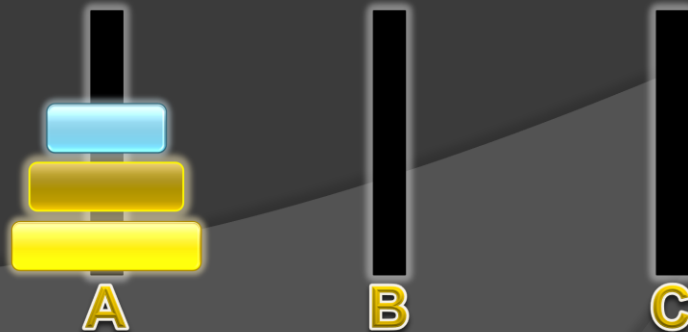
# Towers of Hanoi

```
void solveTowers(counter, source, destination, spare) {  
    if (count == 1)  
        moveDisk(source, destination)  
    else {  
        solveTowers(count - 1, source, spare, destination)  
  
        moveDisk(source, destination)  
  
        solveTowers(count - 1, spare, destination, source)  
    }  
}
```



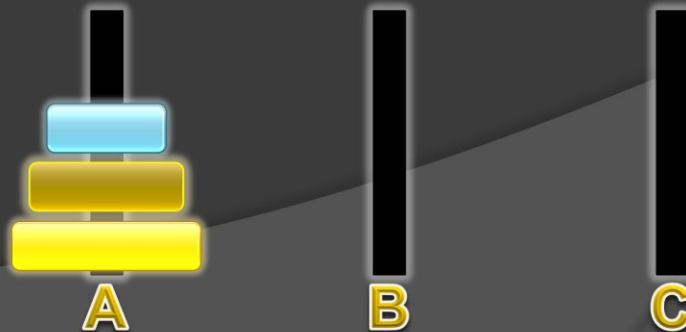
# Towers of Hanoi

```
void solveTowers(3counter, Asource, Cdestination, Bspare) {  
    if (count == 1) AmoveDisk(Csource, destination)  
    else {  
        3-1solveTowers(count - 1, Asource, Bspare, Cdestination) ←  
        moveDisk(source, destination)  
        3-1solveTowers(count - 1, Bspare, Cdestination, Asource)  
    }  
}
```



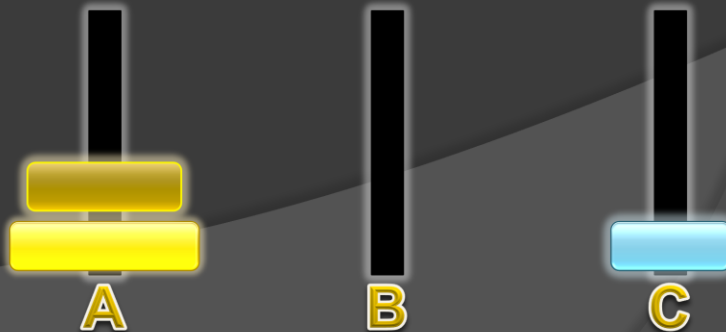
# Towers of Hanoi

```
void solveTowers(2counter, Asource, Bdestination, Cspare) {  
    if (count == 1) AmoveDisk(Bsource, Bdestination)  
    else {  
        2-1solveTowers(count - 1, Asource, Cspare, Bdestination)  
        AmoveDisk(Bsource, Bdestination)  
        2-1solveTowers(count - 1, Cspare, Bdestination, Asource)  
    }  
}
```




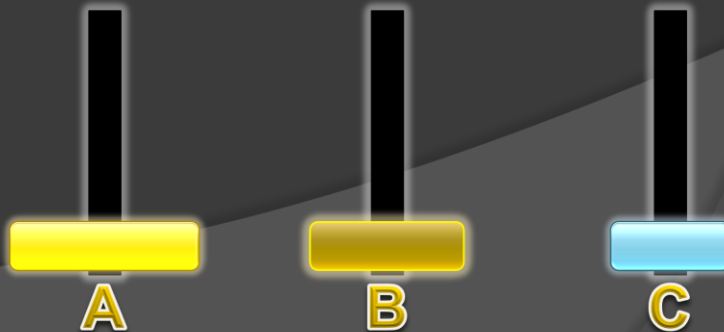
# Towers of Hanoi

```
void solveTowers(1counter, Asource, Cdestination, Bspare) {  
    if (count == 1)  
        moveDisk(Asource, Cdestination) ←  
    else {  
        solveTowers(count - 1, source, spare, destination)  
  
        moveDisk(source, destination)  
  
        solveTowers(count - 1, spare, destination, source)  
    }  
}
```



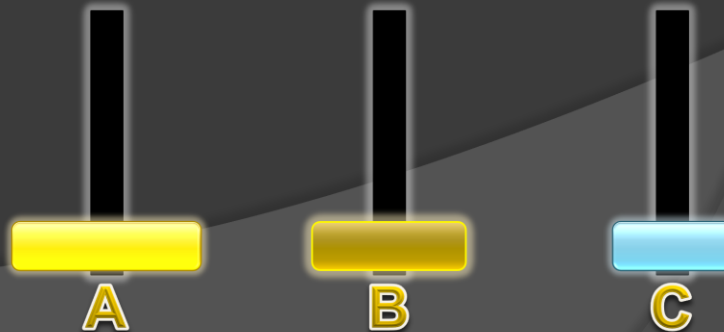
# Towers of Hanoi

```
void solveTowers(2counter, Asource, Bdestination, Cspare) {  
    if (count == 1) AmoveDisk(Bsource, destination)  
    else {  
        2-1solveTowers(count - 1, Asource, Cspare, Bdestination)  
        AmoveDisk(Bsource, destination)   
        2-1solveTowers(count - 1, Cspare, Bdestination, Asource)  
    }  
}
```



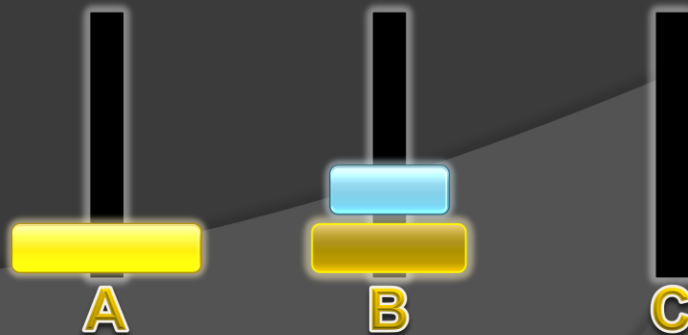
# Towers of Hanoi

```
void solveTowers(2counter, Asource, Bdestination, Cspare) {  
    if (count == 1) AmoveDisk(Bsource, Bdestination)  
    else {  
        2-1solveTowers(count - 1, Asource, Cspare, Bdestination)  
        AmoveDisk(Bsource, Bdestination)  
        2-1solveTowers(count - 1, Cspare, Bdestination, Asource)  
    }  
}
```



# Towers of Hanoi

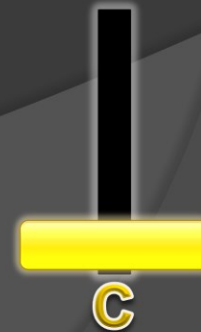
```
void solveTowers(1counter, Csource, Bdestination, Aspare) {  
    if (count == 1) CmoveDisk(Bsource, destination) ←  
    else {  
        solveTowers(count - 1, source, spare, destination)  
  
        moveDisk(source, destination)  
  
        solveTowers(count - 1, spare, destination, source)  
    }  
}
```





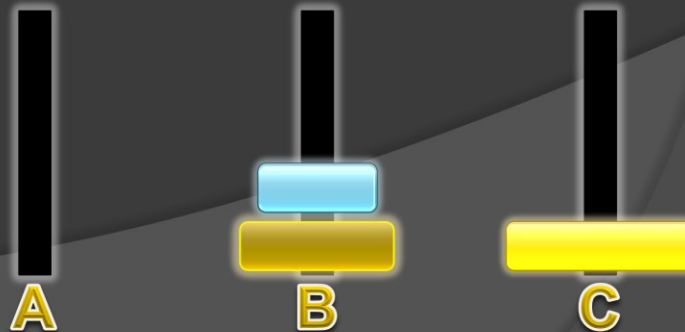
# Towers of Hanoi

```
void solveTowers(3counter, Asource, Cdestination, Bspare) {  
    if (count == 1) AmoveDisk(Csource, destination)  
    else {  
        3-1solveTowers(count - 1, Asource, Bspare, Cdestination)  
        AmoveDisk(source, Cdestination) ←  
        3-1solveTowers(count - 1, Bspare, Cdestination, Asource)  
    }  
}
```



# Towers of Hanoi

```
void solveTowers(3counter, Asource, Cdestination, Bspare) {  
    if (count == 1) AmoveDisk(Csource, destination)  
    else {  
        3-1solveTowers(count - 1, Asource, Bspare, Cdestination)  
        AmoveDisk(Csource, destination)  
        3-1solveTowers(count - 1, Bspare, Cdestination, Asource)  
    }  
}
```



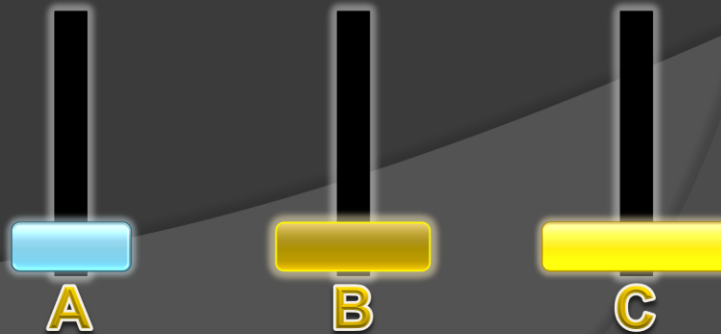
# Towers of Hanoi

```
void solveTowers(2counter, Bsource, Cdestination, Aspare) {  
    if (count == 1) BmoveDisk(Csource, Adestination)  
    else {  
        2-1solveTowers(count - 1, Bsource, Aspare, Cdestination)  
        BmoveDisk(Csource, Adestination)  
        2-1solveTowers(count - 1, Aspare, Cdestination, Bsource)  
    }  
}
```



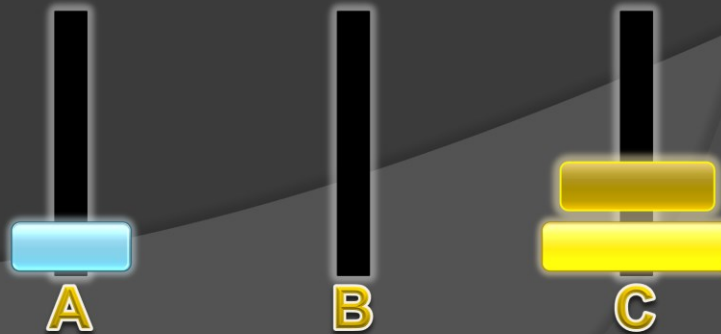
# Towers of Hanoi

```
void solveTowers(1counter, Bsource, Adestination, Cspare) {  
    if (count == 1) BmoveDisk(Asource, destination) ←  
    else {  
        solveTowers(count - 1, source, spare, destination)  
  
        moveDisk(source, destination)  
  
        solveTowers(count - 1, spare, destination, source)  
    }  
}
```



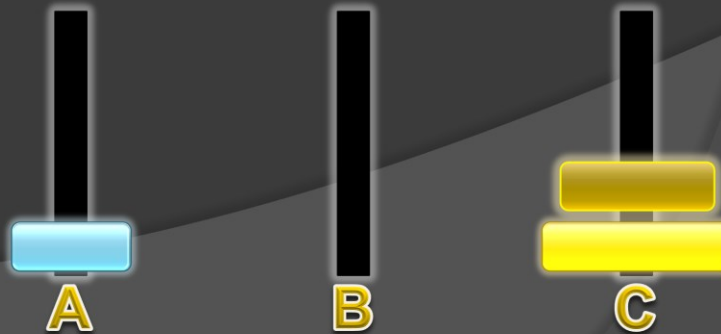
# Towers of Hanoi

```
void solveTowers(2counter, Bsource, Cdestination, Aspare) {  
    if (count == 1) BmoveDisk(Csource, destination)  
    else {  
        2-1solveTowers(count - 1, source, spare, destination)  
        BmoveDisk(Csource, destination) ←  
        2-1solveTowers(count - 1, Aspare, destination, Csource)  
    }  
}
```



# Towers of Hanoi

```
void solveTowers(2counter, Bsource, Cdestination, Aspare) {  
    if (count == 1) BmoveDisk(Csource, destination)  
    else {  
        2-1solveTowers(count - 1, source, spare, destination)  
        BmoveDisk(Csource, destination)  
        2-1solveTowers(count - 1, Aspare, destination, Bsource)  
    }  
}
```



# Towers of Hanoi

```
void solveTowers(1counter, Asource, Cdestination, Bspare) {  
    if (count == 1) AmoveDisk(Csource, destination) ←  
    else {  
        solveTowers(count - 1, source, spare, destination)  
  
        moveDisk(source, destination)  
  
        solveTowers(count - 1, spare, destination, source)  
    }  
}
```

