

Recursion Examples

listLength

listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

main()

current line:

listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

int length

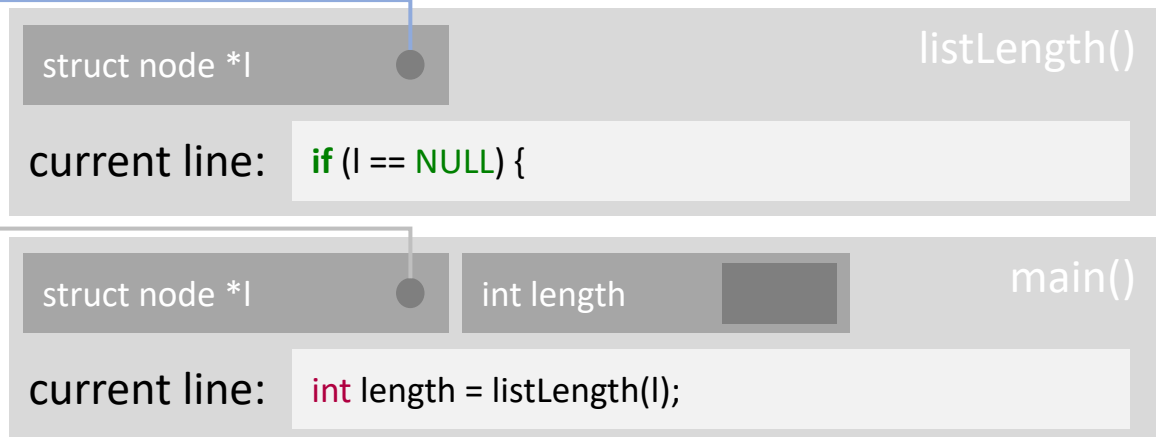
main()

current line: `int length = listLength(l);`

listLength



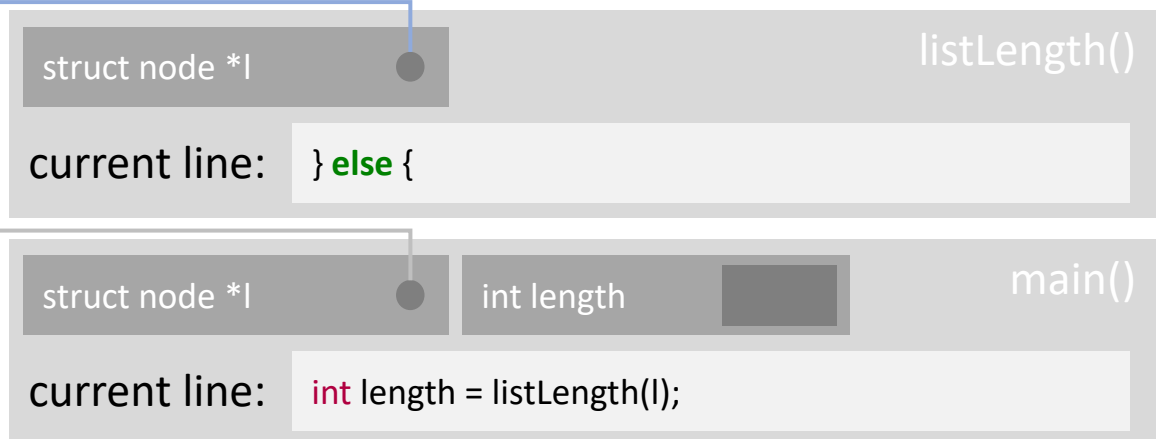
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



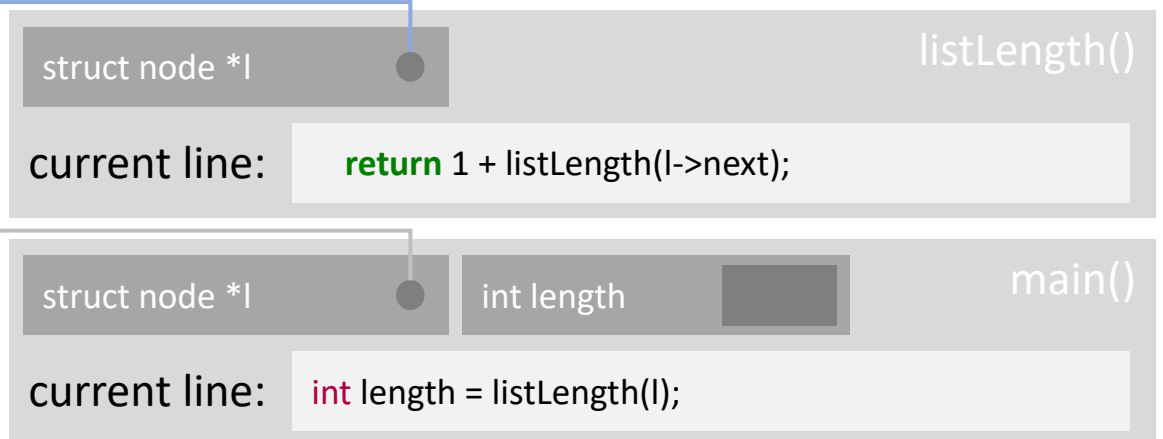
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



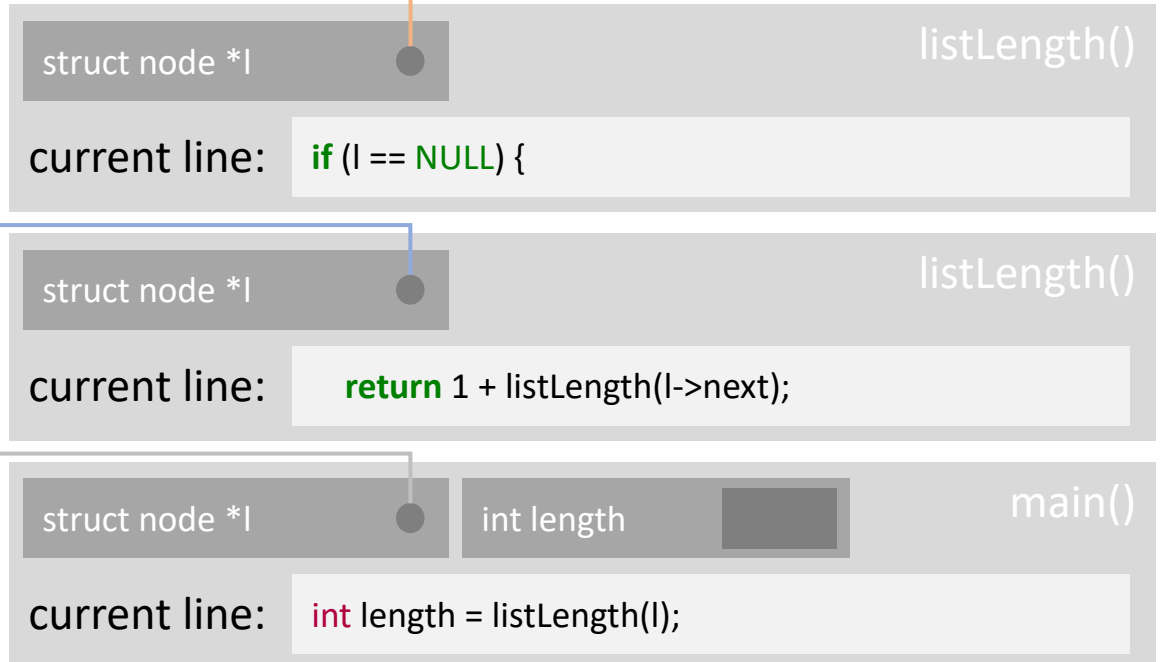
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



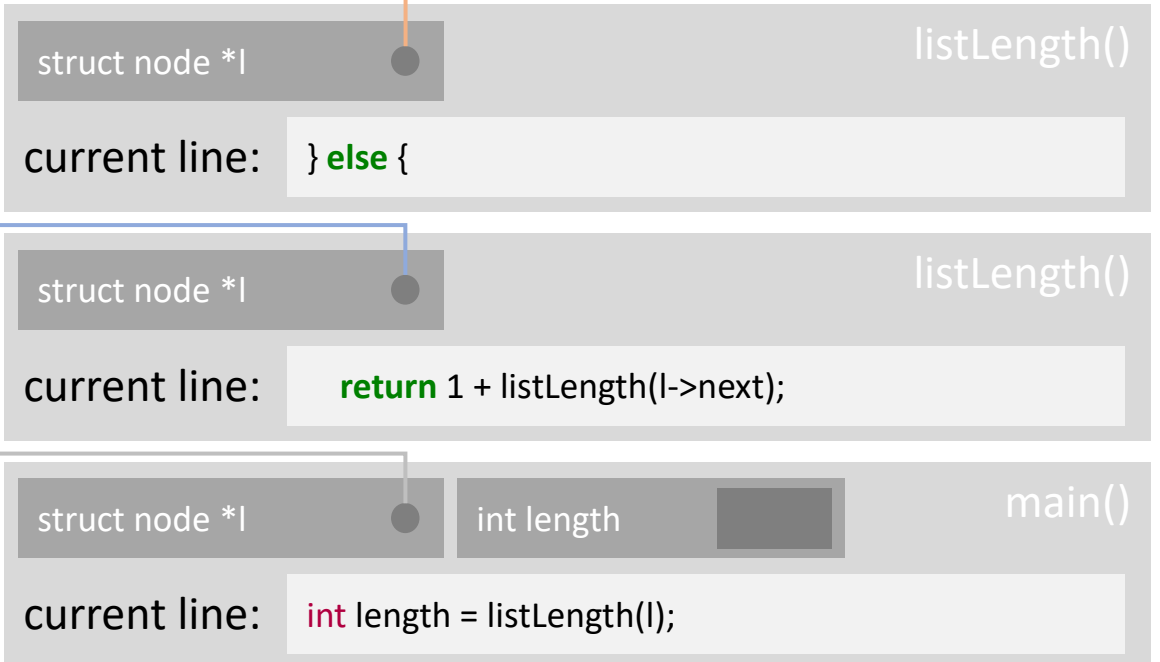
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



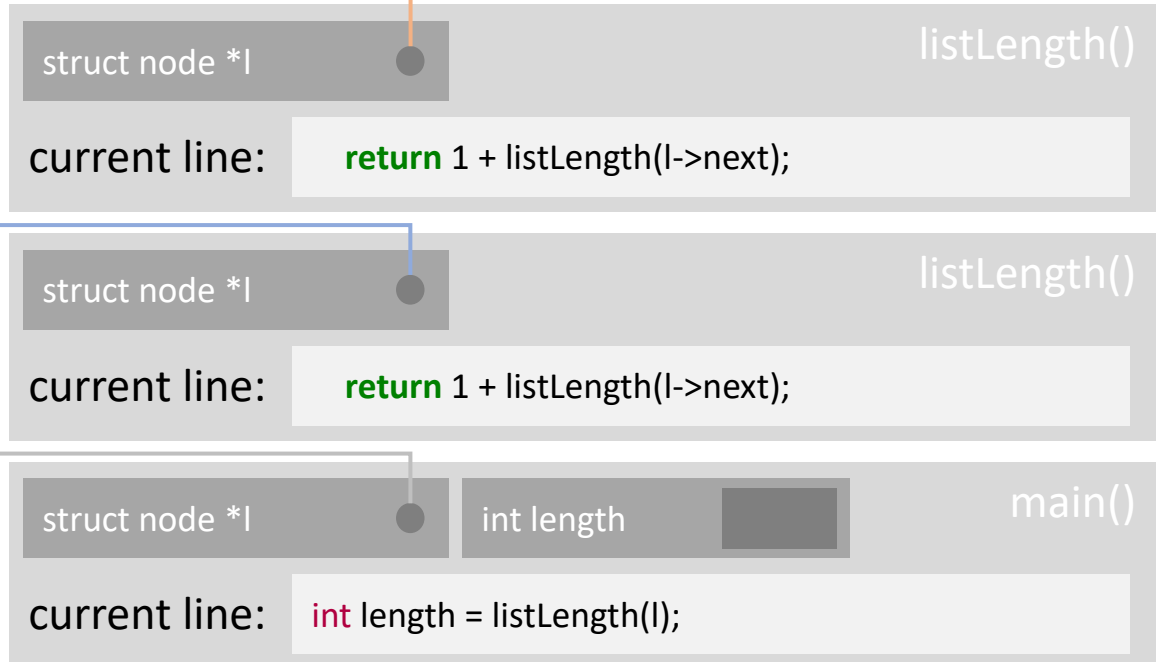
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

current line: `if (l == NULL) {`

listLength()

struct node *l

current line: `return 1 + listLength(l->next);`

listLength()

struct node *l

current line: `return 1 + listLength(l->next);`

listLength()

struct node *l

int length

current line: `int length = listLength(l);`

main()

listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

listLength()

current line: } else {

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

int length

main()

current line: int length = listLength(l);

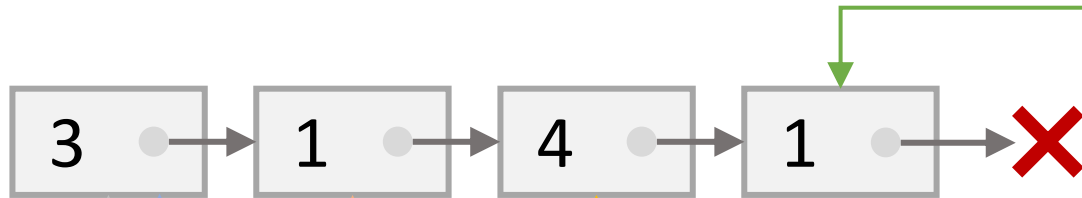
listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

listLength()

current line: if (l == NULL) {

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

listLength()

current line: return 1 + listLength(l->next);

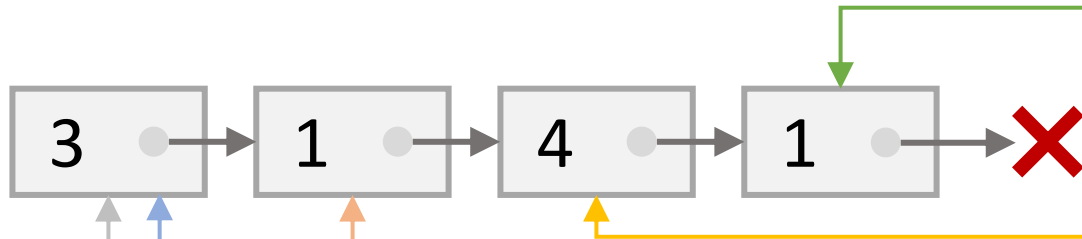
struct node *l

int length

main()

current line: int length = listLength(l);

listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l
current line: } else {
listLength()

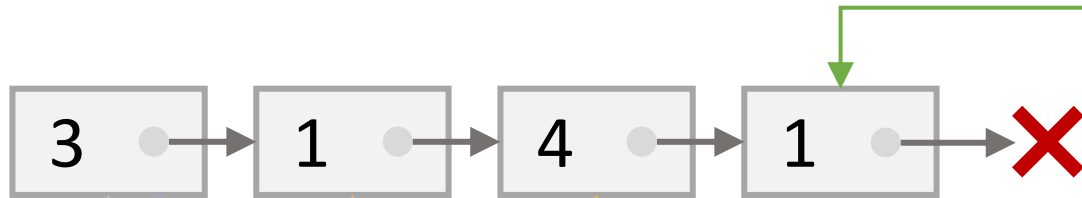
struct node *l
current line: return 1 + listLength(l->next);
listLength()

struct node *l
current line: return 1 + listLength(l->next);
listLength()

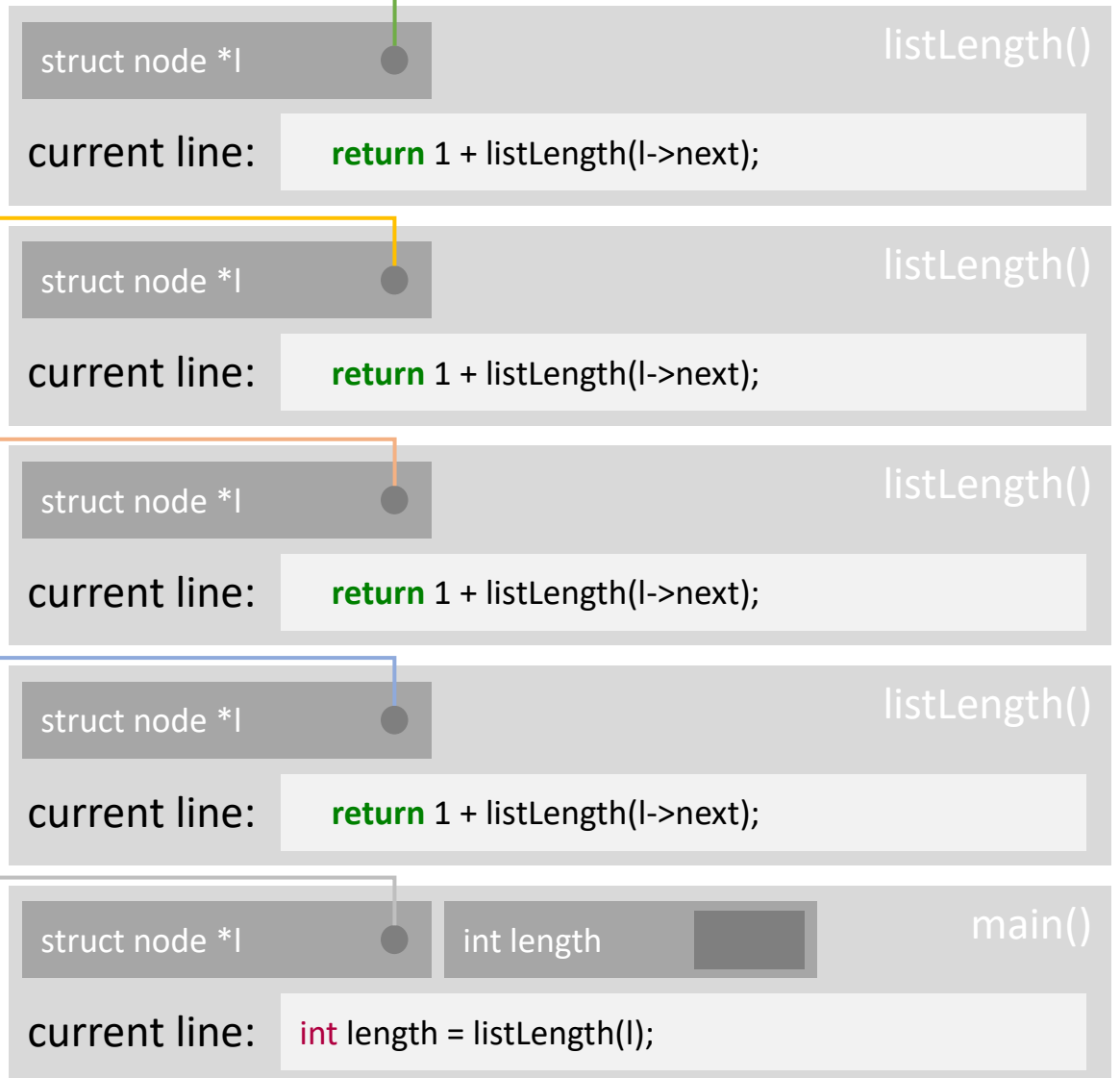
struct node *l
current line: return 1 + listLength(l->next);
listLength()

struct node *l
int length
current line: int length = listLength(l);
main()

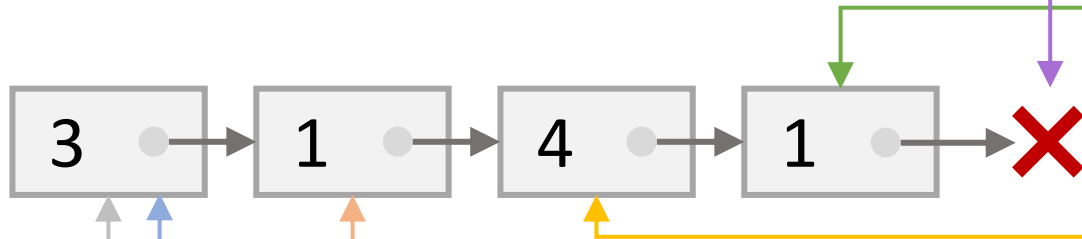
listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

listLength()

current line: if (l == NULL) {

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

listLength()

current line: return 1 + listLength(l->next);

struct node *l

listLength()

current line: return 1 + listLength(l->next);

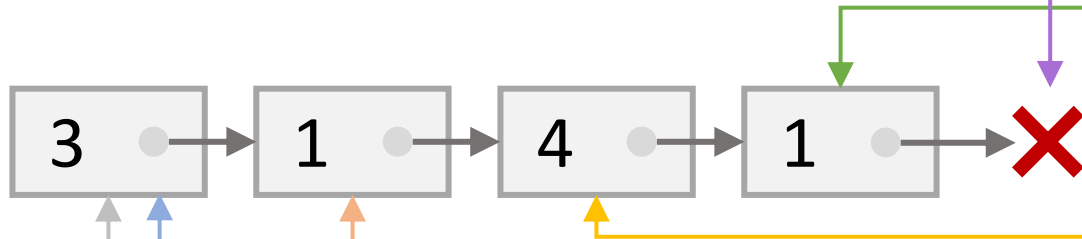
struct node *l

int length

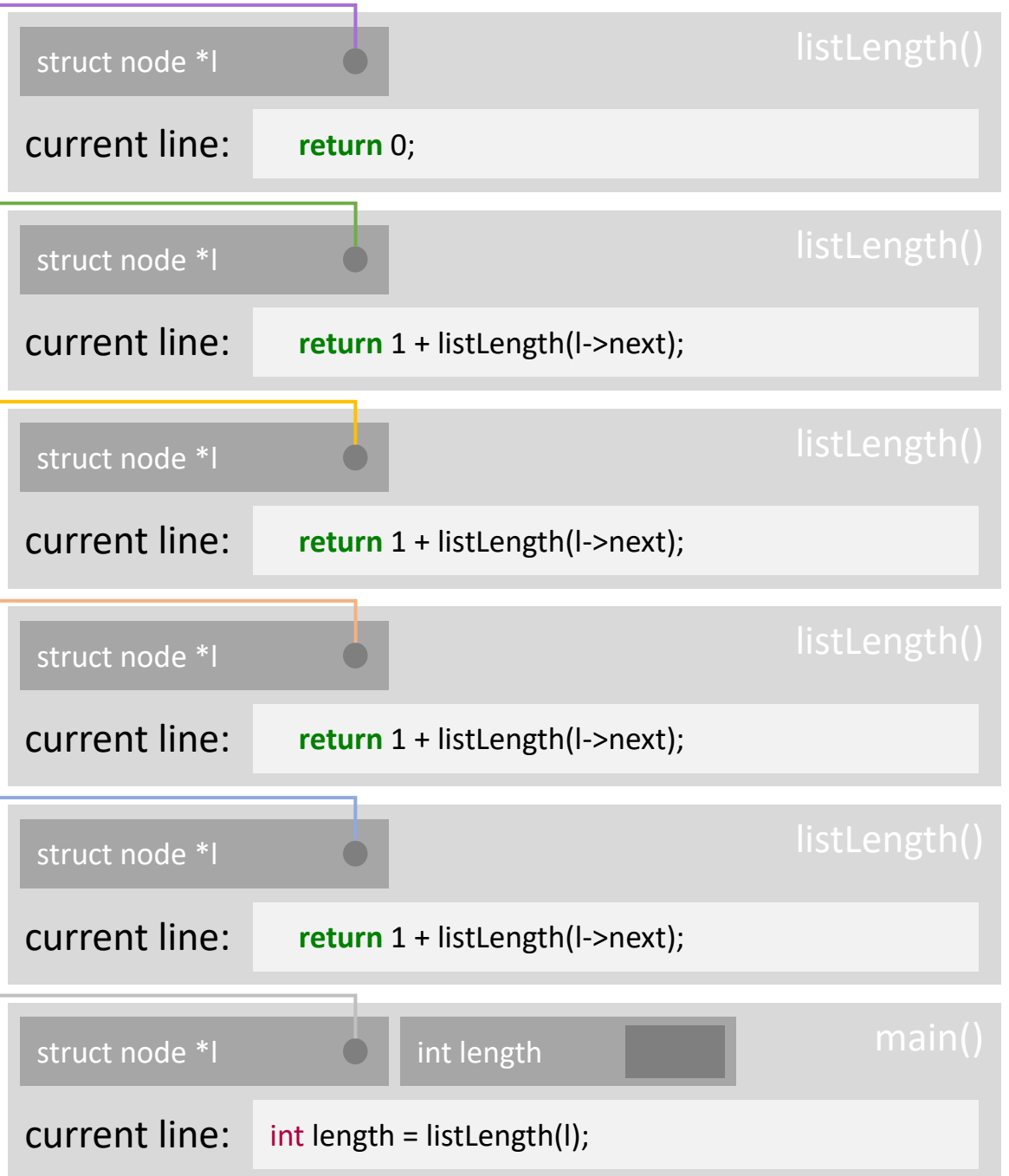
main()

current line: int length = listLength(l);

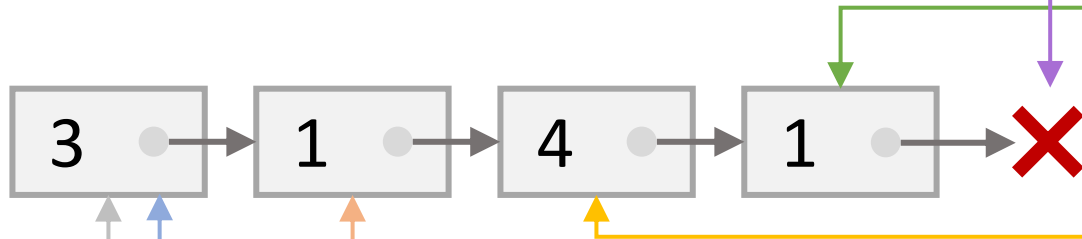
listLength



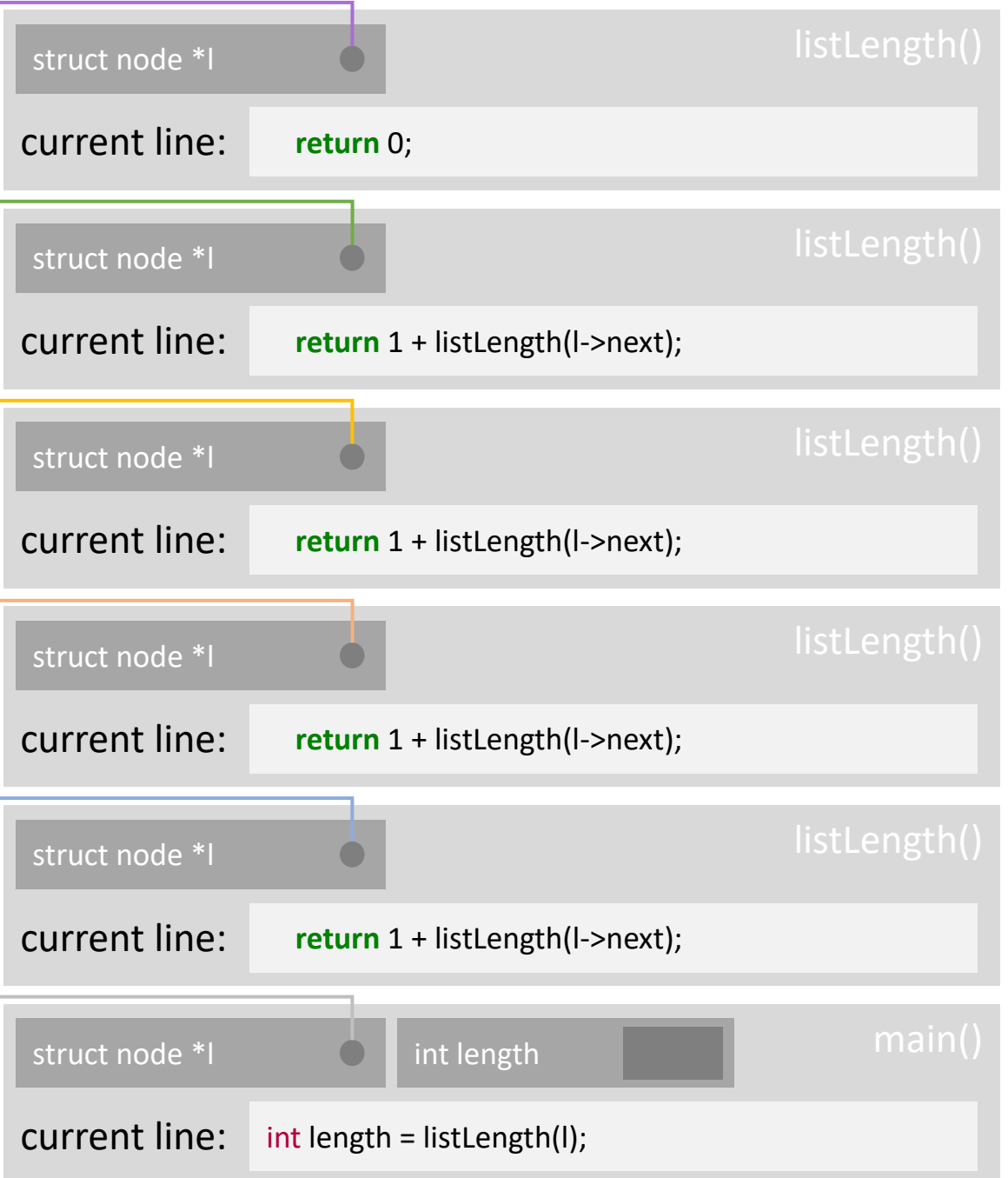
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



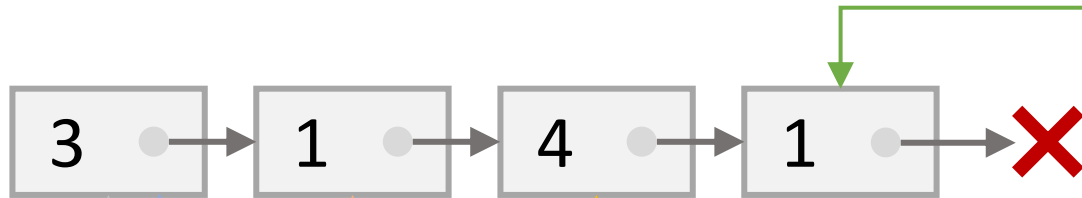
listLength



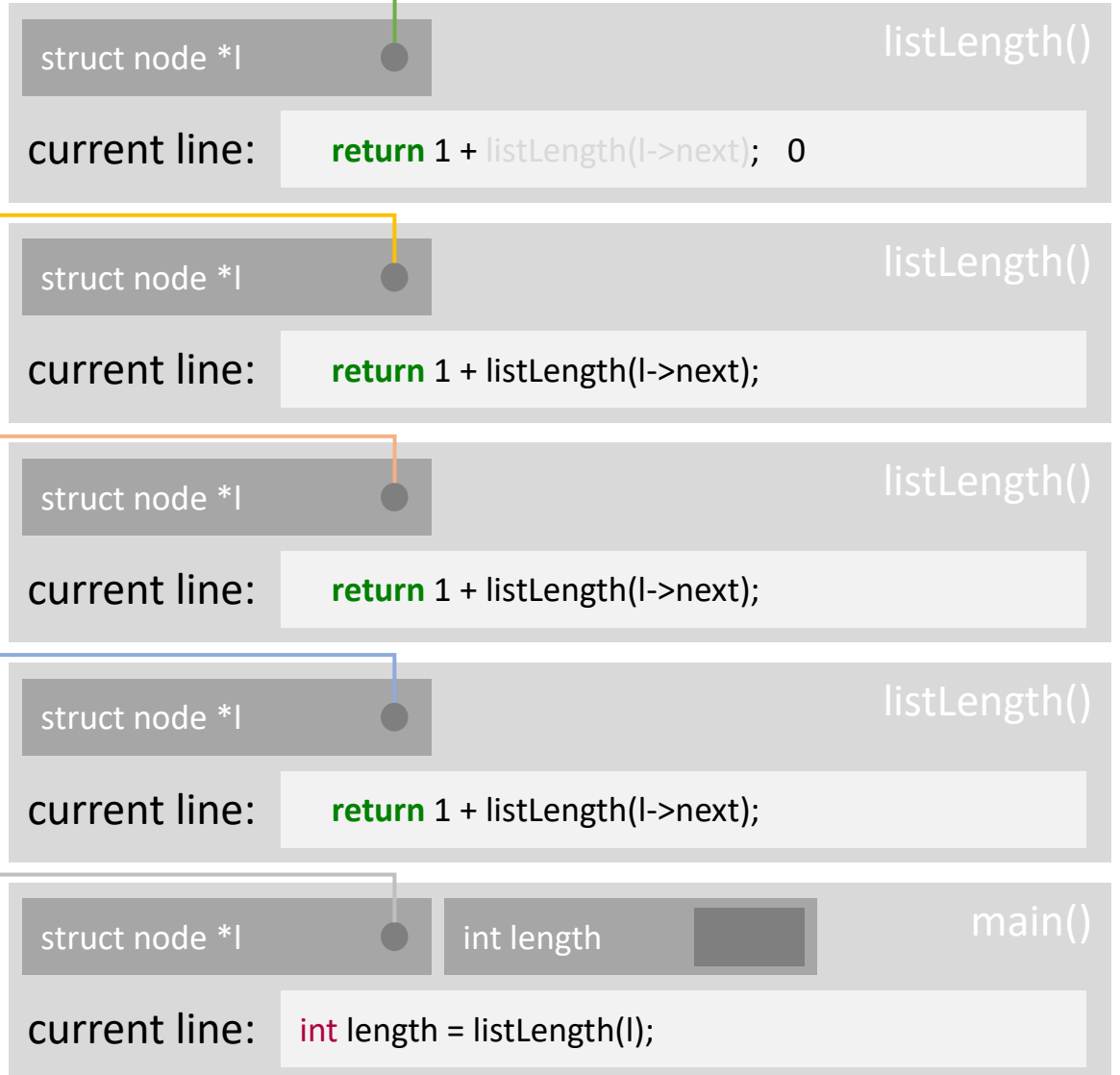
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



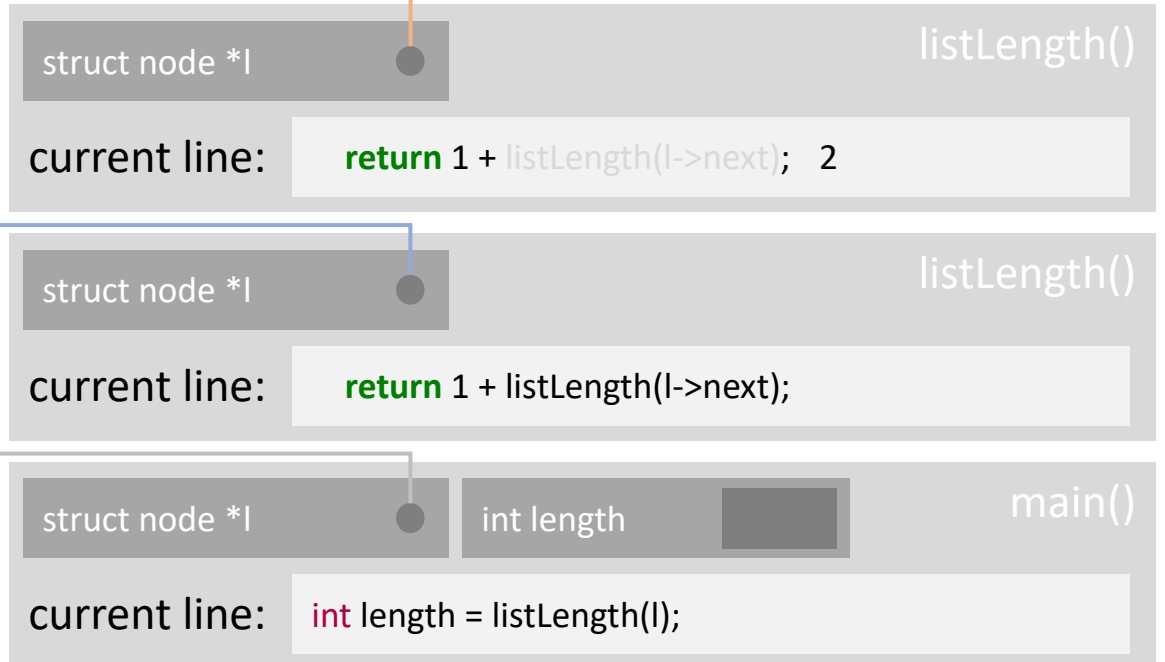
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



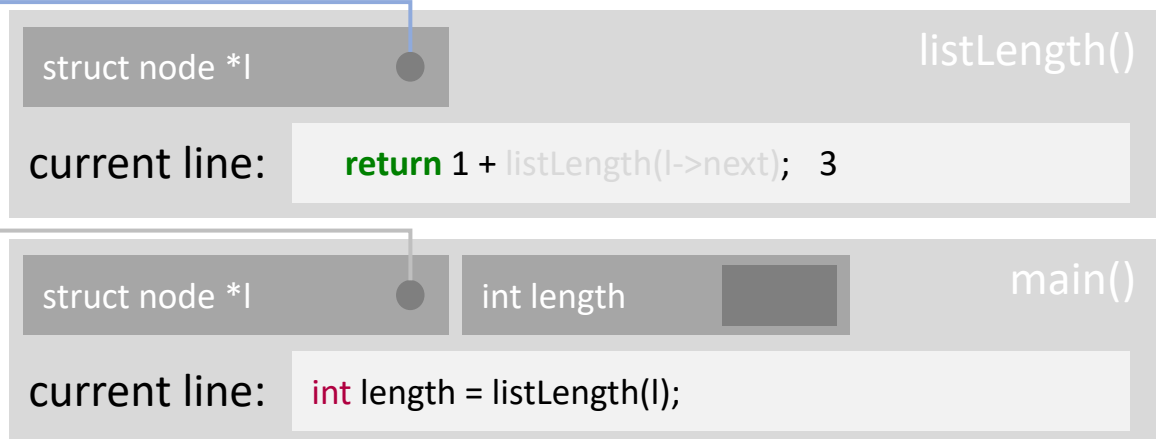
```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```



listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

int length

main()

current line: `int length = listLength(l);` 4

listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

int length

4

main()

current line: `int length = listLength(l);` 4

listLength



```
int main(void) {  
    ...  
    int length = listLength(l);  
}  
  
int listLength(struct node *l) {  
    if (l == NULL) {  
        return 0;  
    } else {  
        return 1 + listLength(l->next);  
    }  
}
```

struct node *l

int length

4

main()

current line: