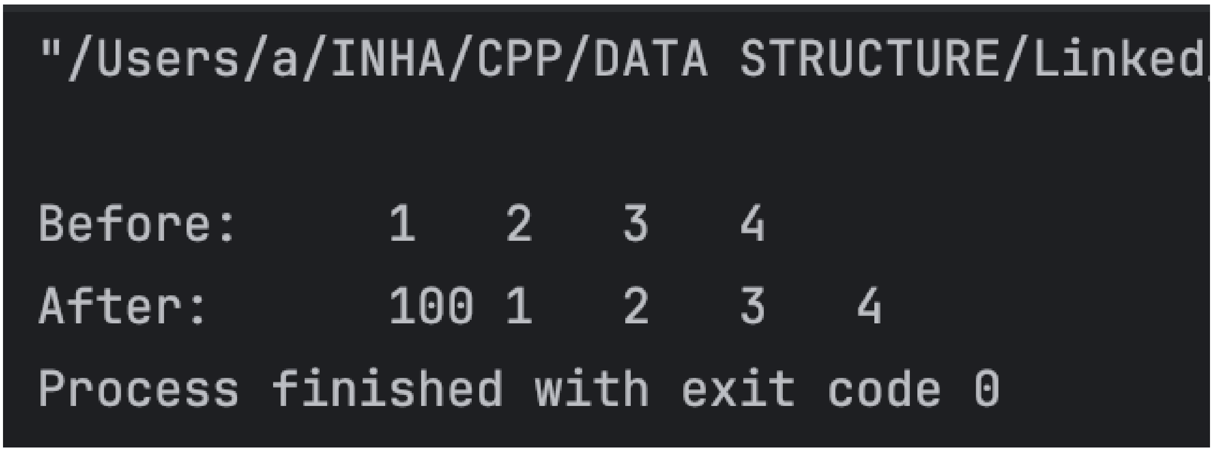
Programming Exercise

ID: 12214760

NAME: YULDOSHEV JAVOKHIR

Insert an element at the beginning of a linked list

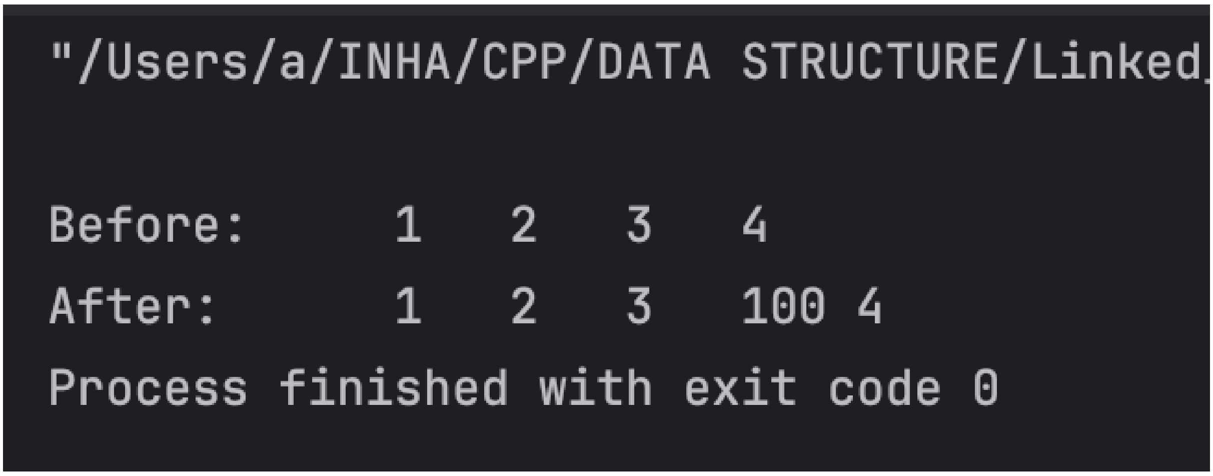
```
#include <iostream>

using namespace  std;
class Node{
public:
    int data;
    Node* nextdata;
    Node* priviousdata;

};

void Insert_beginig( Node** head,Node* first){
    Node * newnode;
    newnode= new Node();
    newnode->data = 100;
    newnode->nextdata = first;
    *head  = newnode;
}


int main(){
    Node* start;
    Node* first ;
    Node* second ;
    Node* third ;
    Node* fourth ;
    first = new Node();
    second = new Node();
    third = new Node();
    fourth = new Node();

    first->data = 1;
    second->data = 2;
    third->data = 3;
    fourth->data = 4;

    first->nextdata = second;
    second->nextdata = third;
    third->nextdata = fourth;
    fourth->nextdata = NULL;

    start = first;
    cout<< "\nBefore: "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    start = first;
    Insert_beginig(&start,first);

    cout<< "\nAfter:   "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    return  0;
}
```



```
"/Users/a/INHA/CPP/DATA STRUCTURE/Linked

Before:     1   2   3   4
After:      100 1   2   3   4
Process finished with exit code 0
```

Insert an element in the middle of a linked list

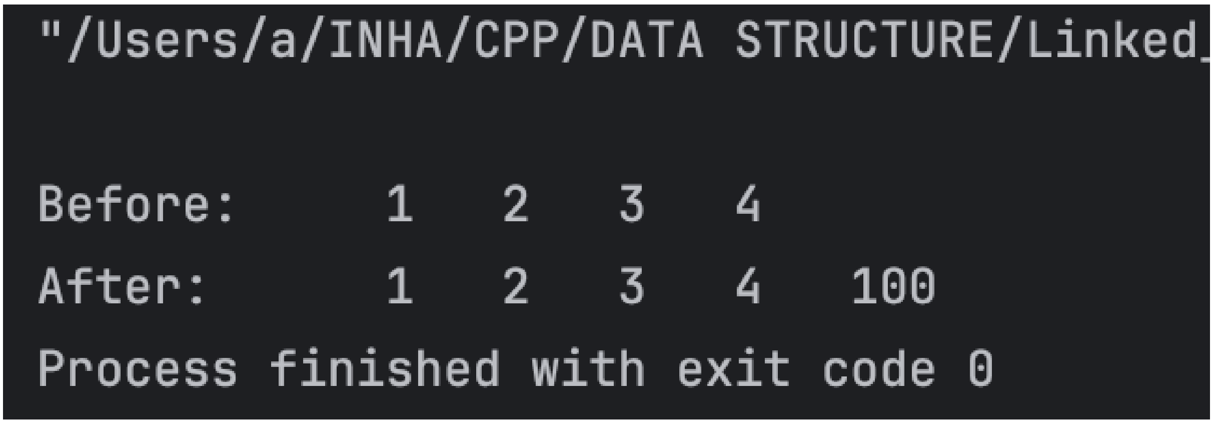
```
#include <iostream>

using namespace  std;
class Node{
public:
    int data;
    Node* nextdata;
    Node* priviousdata;

};


void Insert_mid( Node** head,Node* mid){
    Node * newnode;
    newnode= new Node();
    newnode->data = 100;
    newnode->nextdata = NULL;
    Node* current = *head;
    while (current !=mid){
        current = current->nextdata;
    }
    newnode->nextdata = mid->nextdata;
    current->nextdata = newnode;
}


int main(){
    Node* start;
    Node* first ;
    Node* second ;
    Node* third ;
    Node* fourth ;

    first = new Node();
    second = new Node();
    third = new Node();
    fourth = new Node();

    first->data = 1;
    second->data = 2;
    third->data = 3;
    fourth->data = 4;

    first->nextdata = second;
    second->nextdata = third;
    third->nextdata = fourth;
    fourth->nextdata = NULL;

    start = first;
    cout<< "\nBefore: "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    start = first;
    Insert_mid(&start,third);

    cout<< "\nAfter:   "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    return  0;
}
```



```
"/Users/a/INHA/CPP/DATA STRUCTURE/Linked

Before:     1   2   3   4
After:      1   2   3   100 4
Process finished with exit code 0
```

Insert an element at the end of a linked list

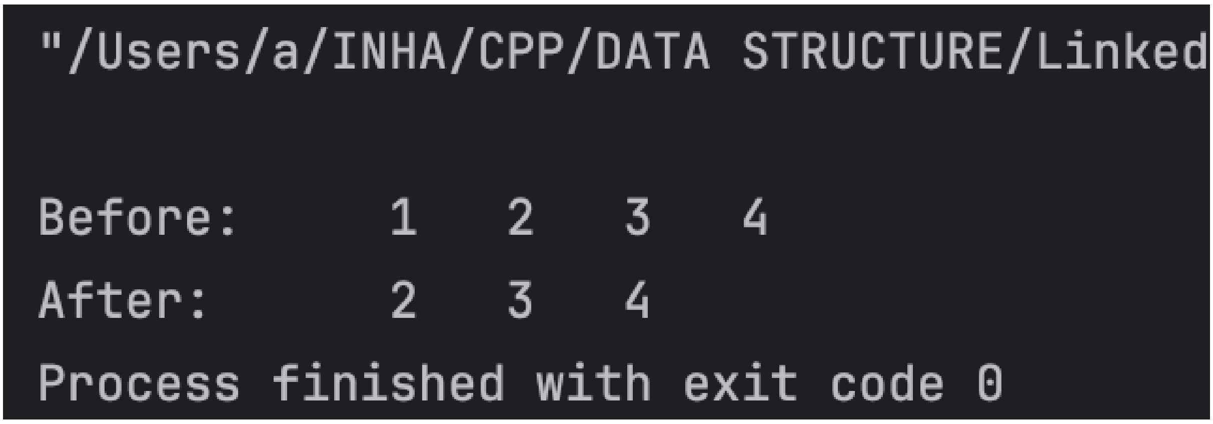
```
#include <iostream>

using namespace  std;
class Node{
public:
    int data;
    Node* nextdata;
    Node* priviousdata;

};


void Insert_end( Node** head){
    Node * newnode;
    newnode= new Node();
    newnode->data = 100;
    newnode->nextdata = NULL;
    Node* current = *head;
    while (current->nextdata !=NULL){
        current = current->nextdata;
    }
    current->nextdata = newnode;
}


int main(){
    Node* start;
    Node* first ;
    Node* second ;
    Node* third ;
    Node* fourth ;

    first = new Node();
    second = new Node();
    third = new Node();
    fourth = new Node();

    first->data = 1;
    second->data = 2;
    third->data = 3;
    fourth->data = 4;

    first->nextdata = second;
    second->nextdata = third;
    third->nextdata = fourth;
    fourth->nextdata = NULL;

    start = first;
    cout<< "\nBefore: "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    start = first;
    Insert_end(&start);
    cout<< "\nAfter:   "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    return  0;
}
```



```
"/Users/a/INHA/CPP/DATA STRUCTURE/Linked_

Before:     1   2   3   4
After:      1   2   3   4   100
Process finished with exit code 0
```

Delete from beginning

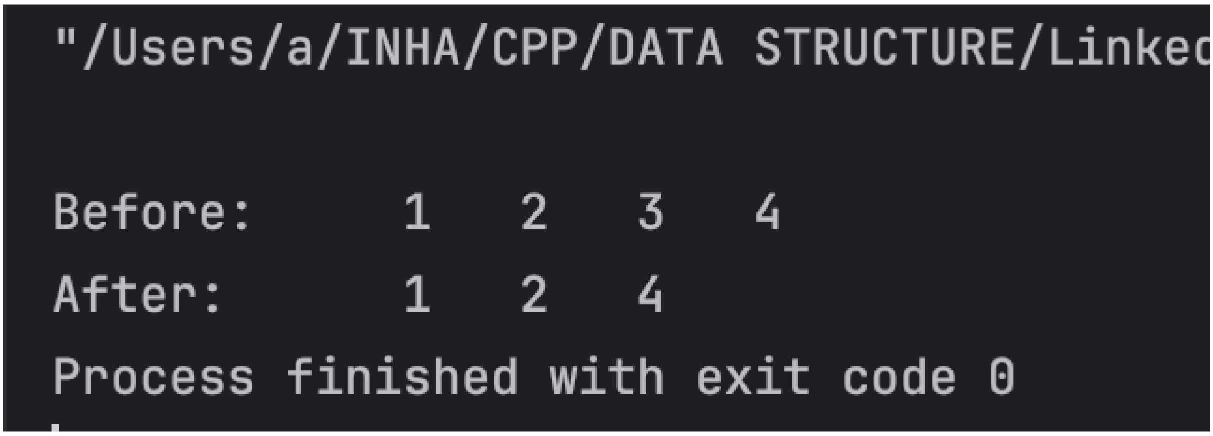
```
#include <iostream>

using namespace  std;
class Node{
public:
    int data;
    Node* nextdata;
    Node* priviousdata;

};



void delete_beginning( Node** head){
    Node* current = *head;
    *head = current->nextdata;

}


int main(){
    Node* start;
    Node* first ;
    Node* second ;
    Node* third ;
    Node* fourth ;

    first = new Node();
    second = new Node();
    third = new Node();
    fourth = new Node();

    first->data = 1;
    second->data = 2;
    third->data = 3;
    fourth->data = 4;

    first->nextdata = second;
    second->nextdata = third;
    third->nextdata = fourth;
    fourth->nextdata = NULL;

    start = first;
    cout<< "\nBefore: "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    start = first;
    delete_beginning(&start);

    cout<< "\nAfter:   "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    return  0;
}
```



```
"/Users/a/INHA/CPP/DATA STRUCTURE/Linked

Before:     1   2   3   4
After:      2   3   4
Process finished with exit code 0
```

Delete from middle

```
#include <iostream>

using namespace  std;
class Node{
public:
    int data;
    Node* nextdata;
    Node* priviousdata;

};


void delete_mid( Node** head, Node* mid){
    Node* current = *head;
    Node* next;
    next = new Node;
    while (current != mid){
        next = current;
        current = current->nextdata;
    }
    next->nextdata = current->nextdata;
}

int main(){
    Node* start;
    Node* first ;
    Node* second ;
    Node* third ;
    Node* fourth ;

    first = new Node();
    second = new Node();
    third = new Node();
    fourth = new Node();

    first->data = 1;
    second->data = 2;
    third->data = 3;
    fourth->data = 4;

    first->nextdata = second;
    second->nextdata = third;
    third->nextdata = fourth;
    fourth->nextdata = NULL;

    start = first;
    cout<< "\nBefore: "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    start = first;
    delete_mid(&start, third);

    cout<< "\nAfter:   "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    return  0;
```

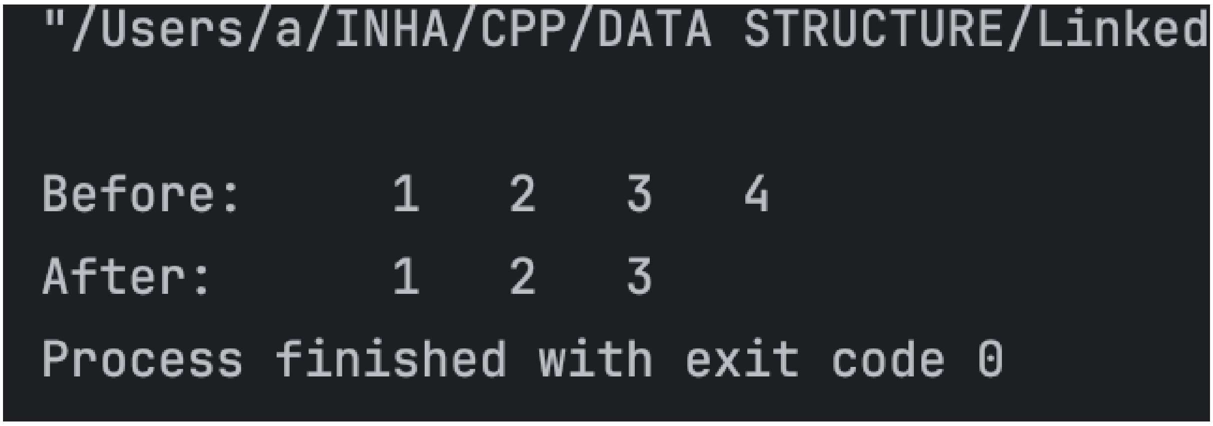


```
"/Users/a/INHA/CPP/DATA STRUCTURE/Linke

Before:      1    2    3    4
After:       1    2    4
Process finished with exit code 0
```

Delete from end

```
#include <iostream>

using namespace  std;
class Node{
public:
    int data;
    Node* nextdata;
    Node* priviousdata;

};


void delete_end( Node** head){
    Node* current = *head;
    while (current->nextdata->nextdata != NULL){
        current = current->nextdata;
    }
    current->nextdata = NULL;
}
```

```
int main(){
    Node* start;
    Node* first ;
    Node* second ;
    Node* third ;
    Node* fourth ;

    first = new Node();
    second = new Node();
    third = new Node();
    fourth = new Node();

    first->data = 1;
    second->data = 2;
    third->data = 3;
    fourth->data = 4;

    first->nextdata = second;
    second->nextdata = third;
    third->nextdata = fourth;
    fourth->nextdata = NULL;

    start = first;
    cout<< "\nBefore: "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    start = first;
    delete_end(&start);


    cout<< "\nAfter:   "<<"\t";
    while(start!= NULL){
        cout << start->data<<"\t";
        start = start->nextdata;
    }
    return  0;
```



```
"/Users/a/INHA/CPP/DATA STRUCTURE/Linked

Before:     1   2   3   4
After:      1   2   3
Process finished with exit code 0
```