

linked list các thứ huhuhuh

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
struct Passenger{
    int age;
    string date; // dd/mm/yyyy
    float weight;
};
struct PaxNode{
    Passenger data;
    PaxNode* next;
};
struct PaxList{
    PaxNode* head;
    PaxNode* tail;
};
void print2D(Passenger** p, int m, int n)
{
    for(int i = 0; i < m; ++i)
    {
        for(int j = 0; j < n; ++j)
        {
            cout << p[i][j].age << " " << p[i][j].date << " " << p[i][j].weight << endl;
        }
    }
}
Passenger* readFile(string filename, int &p, int &m, int &n)
{
    ifstream fin;
    fin.open(filename);
    if(!fin.is_open())
    {
        cout << "cannot read file";
        return nullptr;
    }
    fin >> p >> m >> n;
    fin.ignore();
    string tmp;
    Passenger** a = new Passenger**[p];
    for(int i = 0; i < p; ++i)
```



linked list các thứ huhuhuh (cont)

```

> {
    a[i] = new Passenger*[m];
    for(int j = 0; j < m; ++j)
    {
        a[i][j] = new Passenger[n];
        for(int k = 0; k < n; ++k)
        {
            getline(fin, tmp, ' ');
            a[i][j][k].age = stoi(tmp);
            getline(fin, tmp, ' ');
            a[i][j][k].date = tmp;
            getline(fin, tmp);
            a[i][j][k].weight = stof(tmp);
        }
    }
}
return a;
}

void print3D(Passenger* P, int p, int m, int n)
{
    for(int i = 0; i < p; ++i)
    {
        for(int j = 0; j < m; ++j)
        {
            for(int k = 0; k < n; ++k)
            {
                cout << P[i][j][k].age << " " << P[i][j][k].date << " " << P[i][j][k].weight << endl;
            }
        }
    }
}

Passenger findMinWeight(Passenger* P, int p, int m, int n, int x, int y)
{
    Passenger **ans;
    ans = new Passenger*[p];
    for(int i = 0; i < p; ++i)
    {
        ans[i] = new Passenger[m];
        for(int j = 0; j < m; ++j)
        {
            ans[i][j] = P[i][j][0];
            for(int k = 1; k < n; ++k)

```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 2 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```

>     {
        if(P[i][j][k].weight < ans[i][j].weight)
        {
            ans[i][j] = P[i][j][k];
        }
    }
}
}
return ans;
}
}
PaxNode *createNewNode(Passenger data)
{
    PaxNode *newNode = new PaxNode;
    newNode->data = data;
    newNode->next = nullptr;
    return newNode;
}
PaxList insertSorted(PaxList list, Passenger data)
{
    PaxNode *newNode = createNewNode(data);
    if (list.head == nullptr || list.head->data.age >= newNode->data.age)
    {
        newNode->next = list.head;
        list.head = newNode;
        if(list.tail == nullptr) // If the list was empty, update the tail pointer
        {
            list.tail = newNode;
        }
    }
    else
    {
        PaxNode *current = list.head;
        while (current->next != nullptr && current->next->data.age < newNode->data.age)
        {
            current = current->next;
        }
        newNode->next = current->next;
        current->next = newNode;
        if(current == list.tail) // If the new node was inserted at the end, update the tail pointer
        {
            list.tail = newNode;
        }
    }
}

```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 3 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```

> }
return list;
}
PaxList readFile(string filename)
{
    ifstream fin;
    fin.open(filename);
    if(!fin.is_open())
    {
        cout << "cannot read file";
        return PaxList();
    }
    PaxList list;
    list.head = nullptr;
    list.tail = nullptr;
    string tmp;
    PaxNode *newNode = new PaxNode;
    getline(fin, tmp);
    if(getline(fin, tmp))
    {
        string tmp1 = "";
        int i = 0;
        while(i < tmp.size() && tmp[i] != ' ')
        {
            tmp1 += tmp[i];
            ++i;
        }
        (newNode->data).age = stoi(tmp1);
        tmp1 = "";
        ++i;
        while(i < tmp.size() && tmp[i] != ' ')
        {
            tmp1 += tmp[i];
            ++i;
        }
        (newNode->data).date = tmp1;
        tmp1 = "";
        ++i;
        while(i < tmp.size() && tmp[i] != '\n')
        {
            tmp1 += tmp[i];
            ++i;
        }
    }
}

```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 4 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```

>     }
    (newNode->data).weight = stof(tmp1);
    newNode->next = nullptr;
}
list.head = newNode;
list.tail = newNode;
PaxNode *cur = list.head;
while(getline(fin, tmp))
{
    PaxNode *newNode = new PaxNode;
    string tmp1 = "";
    int i = 0;
    while(i < tmp.size() && tmp[i] != ' ')
    {
        tmp1 += tmp[i];
        ++i;
    }
    (newNode->data).age = stoi(tmp1);
    tmp1 = "";
    ++i;
    while(i < tmp.size() && tmp[i] != ' ')
    {
        tmp1 += tmp[i];
        ++i;
    }
    (newNode->data).date = tmp1;
    tmp1 = "";
    ++i;
    while(i < tmp.size() && tmp[i] != '\n')
    {
        tmp1 += tmp[i];
        ++i;
    }
    (newNode->data).weight = stof(tmp1);
    list = insertSorted(list, newNode->data);
}
return list;
}
void printList(PaxList list)
{
    PaxNode *cur = list.head;
    while(cur != nullptr)

```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 5 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> {
    cout << cur->data.age << " " << cur->data.date << " " << cur->data.weight << endl;
    cur = cur->next;
}
}
void removePassenger(PaxList &list)
{
    while(list.head != nullptr && list.head->next != nullptr && list.head->data.age % 2 == 0 && list.head->next->data.age % 2 == 0)
    {
        PaxNode *tmp = list.head;
        list.head = list.head->next;
        delete tmp;
    }
    if(list.head == nullptr || list.head->next == nullptr)
    {
        return;
    }
    PaxNode *cur1 = list.head;
    while (cur1->next != nullptr)
    {
        PaxNode *prev = cur1;
        PaxNode *cur2 = cur1->next;
        while(cur2 != nullptr)
        {
            if ((cur1->data).age % 2 == 0 && (cur2->data).age % 2 == 0)
            {
                PaxNode *tmp = cur2;
                cur2 = cur2->next;
                prev->next = cur2;
                delete tmp;
            }
            prev = cur2;
            cur2 = cur2->next;
        }
        cur1 = cur1->next;
    }
}
int main()
{
    string doc = "train.txt";
    // int p, m, n;
    // Passenger *P = readFile(doc, p, m, n);
```

By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 6 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```

> // print3D(P, p, m, n);
// cout << "-----" << endl;
// Passenger **minWeight = findMinWeight(P, p, m, n, 2, 2);
// print2D(minWeight, p, m);
PaxList list = readFile(doc);
printList(list);
removePassenger(list);
cout << "-----" << endl;
printList(list);
return 0;
}
void stackPush(Train a, List &stack)
{
    Node *tmp = stack.pHead;
    Node *newNode = createNewNode(a);
    if(stack.pHead == nullptr && stack.pTail == nullptr)
    {
        stack.pHead = newNode;
        stack.pTail = newNode;
        return;
    }
    newNode->next = stack.pHead;
    stack.pTail = stack.pHead;
    stack.pHead = newNode;
}
bool stackPop(List &stack, Train &x)
{
    if(stack.pHead == nullptr)
    {
        return false;
    }
    Node *tmp = stack.pHead;
    stack.pHead = stack.pHead->next;
    x = tmp->data;
    delete tmp;
    return true;
}
void endQ(Train a, List &queue)
{
    Node *newNode = createNewNode(a);
    if(queue.pHead == nullptr)
    {

```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 7 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
>   queue.pHead = newNode;
    queue.pTail = newNode;
    return;
}
queue.pTail->next = newNode;
queue.pTail = newNode;
}
bool deQ(Train &a, List &queue)
{
    if(queue.pHead == nullptr)
    {
        return false;
    }
    Node *tmp = queue.pHead;
    queue.pHead = queue.pHead->next;
    a = tmp->data;
    delete tmp;
    return true;
}
#include <iostream>
using namespace std;
//create a node
struct Node{
    int data;
    Node *next;
};
struct List{
    Node *head;
    Node *tail;
};
Node *createANode(int x)
{
    Node *newNode = new Node;
    newNode->data = x;
    newNode->next = nullptr;
    return newNode;
}
void addToHead1(int x, List &list)
{
    Node *tmp = list.head;
    Node *newNode = createANode(x);
    if(list.head == nullptr)
```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 8 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> {
    list.head = newNode;
    list.tail = newNode;
    return;
}
newNode->next = list.head;
list.head = newNode;
}
void addToTail1(int x, List &list)
{
    Node *newNode = createANode(x);
    if(list.head == nullptr)
    {
        list.head = newNode;
        list.tail = newNode;
    }
    else
    {
        list.tail->next = newNode;
        list.tail = newNode;
    }
}
void insertNewNodeBeforeK(int x, int k, List &list)
{
    Node *newNode = createANode(x);
    Node *tmp1 = list.head;
    if(list.head == nullptr)
    {
        return;
    }
    if(tmp1->data == k)
    {
        addToHead1(x, list);
        return;
    }
    Node *tmp2 = tmp1->next;
    while(tmp1 != nullptr && tmp2->data != k && tmp2 != nullptr)
    {
        tmp1 = tmp1->next;
        tmp2 = tmp1->next;
    }
    tmp1->next = newNode;
```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 9 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> newNode->next = tmp2;
}
void deleteHead1(List list)
{
    if(list.head == nullptr)
    {
        return;
    }
    Node *tmp = list.head;
    delete tmp;
}
void deleteTail1(List &list)
{
    if(list.head == nullptr)
    {
        return;
    }
    Node *tmp1 = list.head;
    while(tmp1->next != list.tail)
    {
        tmp1 = tmp1->next;
    }
    Node *tmp = list.tail;
    list.tail = tmp1;
    list.tail->next = nullptr;
    delete tmp;
}
void deleteRandomly(List &list, int k)
{
    if(list.head == nullptr)
    {
        return;
    }
    if(list.head->data == k)
    {
        Node *tmp = list.head;
        list.head = list.head->next;
        delete tmp;
        return;
    }
    Node *tmp1 = list.head;
    Node *tmp2 = tmp1->next;
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 10 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> while(tmp1 != nullptr && tmp2->data != k)
{
    tmp1 = tmp1->next;
    tmp2 = tmp1;
}
if(tmp2 != nullptr)
{
    tmp1->next = tmp2->next;
    delete tmp2;
}
}
void printList1(List list)
{
    while(list.head != nullptr)
    {
        cout << list.head->data << " ";
        list.head = list.head->next;
    }
}
void printList2(Node *tmp)
{
    if(tmp != nullptr)
    {
        cout << tmp->data << " ";
        printList2(tmp->next);
    }
}
//trivial
#include "3_2.h"
node *createANode(int x)
{
    node *NewNode = new node;
    NewNode->data = x;
    NewNode->next = nullptr;
    return NewNode;
}
void loadFile(ifstream &input, LinkedList &list)
{
    list.head = nullptr;
    int data;
    input >> data;
    if (data == 0)
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 11 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> {
    return;
}
node *NewNode = createANode(data);
list.head = NewNode;
node *tail = list.head;
while (true)
{
    input >> data;
    if (data == 0)
    {
        break;
    }
    NewNode = createANode(data);
    tail->next = NewNode;
    tail = tail->next;
}
}
```

linked list các thứ huhuhuh

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
struct Passenger{
    int age;
    string date; // dd/mm/yyyy
    float weight;
};
struct PaxNode{
    Passenger data;
    PaxNode* next;
};
struct PaxList{
    PaxNode* head;
    PaxNode* tail;
};
void print2D(Passenger** p, int m, int n)
{
    for(int i = 0; i < m; ++i)
    {
        for(int j = 0; j < n; ++j)
        {
```



linked list các thứ huhuhuh (cont)

```
>     cout << p[i][j].age << " " << p[i][j].date << " " << p[i][j].weight << endl;
    }
    }
}
Passenger *readFile(string filename, int &p, int &m, int &n)
{
    ifstream fin;
    fin.open(filename);
    if(!fin.is_open())
    {
        cout << "cannot read file";
        return nullptr;
    }
    fin >> p >> m >> n;
    fin.ignore();
    string tmp;
    Passenger *a;
    a = new Passenger** [p];
    for(int i = 0; i < p; ++i)
    {
        a[i] = new Passenger*[m];
        for(int j = 0; j < m; ++j)
        {
            a[i][j] = new Passenger[n];
            for(int k = 0; k < n; ++k)
            {
                getline(fin, tmp, ' ');
                a[i][j][k].age = stoi(tmp);
                getline(fin, tmp, ' ');
                a[i][j][k].date = tmp;
                getline(fin, tmp);
                a[i][j][k].weight = stof(tmp);
            }
        }
    }
    return a;
}
void print3D(Passenger**P, int p, int m, int n)
{
    for(int i = 0; i < p; ++i)
    {
        for(int j = 0; j < m; ++j)
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 13 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> {
    for(int k = 0; k < n; ++k)
    {
        cout << P[i][j][k].age << " " << P[i][j][k].date << " " << P[i][j][k].weight << endl;
    }

}
}
}
Passenger findMinWeight(Passenger* P, int p, int m, int n, int x, int y)
{
    Passenger **ans;
    ans = new Passenger*[p];
    for(int i = 0; i < p; ++i)
    {
        ans[i] = new Passenger[m];
        for(int j = 0; j < m; ++j)
        {
            ans[i][j] = P[i][j][0];
            for(int k = 1; k < n; ++k)
            {
                if(P[i][j][k].weight < ans[i][j].weight)
                {
                    ans[i][j] = P[i][j][k];
                }
            }
        }
    }
    return ans;
}
PaxNode *createNewNode(Passenger data)
{
    PaxNode *newNode = new PaxNode;
    newNode->data = data;
    newNode->next = nullptr;
    return newNode;
}
PaxList insertSorted(PaxList list, Passenger data)
{
    PaxNode *newNode = createNewNode(data);
    if (list.head == nullptr || list.head->data.age >= newNode->data.age)
    {
        newNode->next = list.head;
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 14 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> list.head = newNode;
if(list.tail == nullptr) // If the list was empty, update the tail pointer
{
    list.tail = newNode;
}
}
else
{
    PaxNode *current = list.head;
    while (current->next != nullptr && current->next->data.age < newNode->data.age)
    {
        current = current->next;
    }
    newNode->next = current->next;
    current->next = newNode;
    if(current == list.tail) // If the new node was inserted at the end, update the tail pointer
    {
        list.tail = newNode;
    }
}
return list;
}
PaxList readFile(string filename)
{
    ifstream fin;
    fin.open(filename);
    if(!fin.is_open())
    {
        cout << "cannot read file";
        return PaxList();
    }
    PaxList list;
    list.head = nullptr;
    list.tail = nullptr;
    string tmp;
    PaxNode *newNode = new PaxNode;
    getline(fin, tmp);
    if(getline(fin, tmp))
    {
        string tmp1 = "";
        int i = 0;
        while(i < tmp.size() && tmp[i] != ' ')
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 15 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> {
    tmp1 += tmp[i];
    ++i;
}
(newNode->data).age = stoi(tmp1);
tmp1 = "";
++;
while(i < tmp.size() && tmp[i] != ' ')
{
    tmp1 += tmp[i];
    ++i;
}
(newNode->data).date = tmp1;
tmp1 = "";
++;
while(i < tmp.size() && tmp[i] != '\n')
{
    tmp1 += tmp[i];
    ++i;
}
(newNode->data).weight = stof(tmp1);
newNode->next = nullptr;
}
list.head = newNode;
list.tail = newNode;
PaxNode *cur = list.head;
while(getline(fin, tmp))
{
    PaxNode *newNode = new PaxNode;
    string tmp1 = "";
    int i = 0;
    while(i < tmp.size() && tmp[i] != ' ')
    {
        tmp1 += tmp[i];
        ++i;
    }
    (newNode->data).age = stoi(tmp1);
    tmp1 = "";
    ++i;
    while(i < tmp.size() && tmp[i] != ' ')
    {
        tmp1 += tmp[i];
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 16 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```

>     ++i;
    }
    (newNode->data).date = tmp1;
    tmp1 = "";
    ++i;
    while(i < tmp.size() && tmp[i] != '\n')
    {
        tmp1 += tmp[i];
        ++i;
    }
    (newNode->data).weight = stof(tmp1);
    list = insertSorted(list, newNode->data);
}
return list;
}
void printList(PaxList list)
{
    PaxNode *cur = list.head;
    while(cur != nullptr)
    {
        cout << cur->data.age << " " << cur->data.date << " " << cur->data.weight << endl;
        cur = cur->next;
    }
}
void removePassenger(PaxList &list)
{
    while(list.head != nullptr && list.head->next != nullptr && list.head->data.age % 2 == 0 && list.head->next->data.age % 2 == 0)
    {
        PaxNode *tmp = list.head;
        list.head = list.head->next;
        delete tmp;
    }
    if(list.head == nullptr || list.head->next == nullptr)
    {
        return;
    }
    PaxNode *cur1 = list.head;
    while (cur1->next != nullptr)
    {
        PaxNode *prev = cur1;
        PaxNode *cur2 = cur1->next;
        while(cur2 != nullptr)

```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 17 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```

> {
    if ((cur1->data).age % 2 == 0 && (cur2->data).age % 2 == 0)
    {
        PaxNode *tmp = cur2;
        cur2 = cur2->next;
        prev->next = cur2;
        delete tmp;
    }
    prev = cur2;
    cur2 = cur2->next;
}
cur1 = cur1->next;
}
}
int main()
{
    string doc = "train.txt";
    // int p, m, n;
    // Passenger P = readFile(doc, p, m, n);
    // print3D(P, p, m, n);
    // cout << "-----" << endl;
    // Passenger **minWeight = findMinWeight(P, p, m, n, 2, 2);
    // print2D(minWeight, p, m);
    PaxList list = readFile(doc);
    printList(list);
    removePassenger(list);
    cout << "-----" << endl;
    printList(list);
    return 0;
}
void stackPush(Train a, List &stack)
{
    Node *tmp = stack.pHead;
    Node *newNode = createNewNode(a);
    if(stack.pHead == nullptr && stack.pTail == nullptr)
    {
        stack.pHead = newNode;
        stack.pTail = newNode;
        return;
    }
    newNode->next = stack.pHead;
    stack.pTail = stack.pHead;
}

```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 18 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> stack.pHead = newNode;
}
bool stackPop(List &stack, Train &x)
{
    if(stack.pHead == nullptr)
    {
        return false;
    }
    Node *tmp = stack.pHead;
    stack.pHead = stack.pHead->next;
    x = tmp->data;
    delete tmp;
    return true;
}
void endQ(Train a, List &queue)
{
    Node *newNode = createNewNode(a);
    if(queue.pHead == nullptr)
    {
        queue.pHead = newNode;
        queue.pTail = newNode;
        return;
    }
    queue.pTail->next = newNode;
    queue.pTail = newNode;
}
bool deQ(Train &a, List &queue)
{
    if(queue.pHead == nullptr)
    {
        return false;
    }
    Node *tmp = queue.pHead;
    queue.pHead = queue.pHead->next;
    a = tmp->data;
    delete tmp;
    return true;
}
#include <iostream>
using namespace std;
//create a node
struct Node{
```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 19 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> int data;
   Node *next;
};
struct List{
   Node *head;
   Node *tail;
};
Node *createANode(int x)
{
   Node *newNode = new Node;
   newNode->data = x;
   newNode->next = nullptr;
   return newNode;
}
void addToHead1(int x, List &list)
{
   Node *tmp = list.head;
   Node *newNode = createANode(x);
   if(list.head == nullptr)
   {
      list.head = newNode;
      list.tail = newNode;
      return;
   }
   newNode->next = list.head;
   list.head = newNode;
}
void addToTail1(int x, List &list)
{
   Node *newNode = createANode(x);
   if(list.head == nullptr)
   {
      list.head = newNode;
      list.tail = newNode;
   }
   else
   {
      list.tail->next = newNode;
      list.tail = newNode;
   }
}
void insertNewNodeBeforeK(int x, int k, List &list)
```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 20 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> {
    Node *newNode = createANode(x);
    Node *tmp1 = list.head;
    if(list.head == nullptr)
    {
        return;
    }
    if(tmp1->data == k)
    {
        addToHead1(x, list);
        return;
    }
    Node *tmp2 = tmp1->next;
    while(tmp1 != nullptr && tmp2->data != k && tmp2 != nullptr)
    {
        tmp1 = tmp1->next;
        tmp2 = tmp1->next;
    }
    tmp1->next = newNode;
    newNode->next = tmp2;
}
void deleteHead1(List list)
{
    if(list.head == nullptr)
    {
        return;
    }
    Node *tmp = list.head;
    delete tmp;
}
void deleteTail1(List &list)
{
    if(list.head == nullptr)
    {
        return;
    }
    Node *tmp1 = list.head;
    while(tmp1->next != list.tail)
    {
        tmp1 = tmp1->next;
    }
    Node *tmp = list.tail;
```



By **emilyle1104**

Not published yet.

Last updated 7th June, 2024.

Page 21 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> list.tail = tmp1;
list.tail->next = nullptr;
delete tmp;
}
void deleteRandomly(List &list, int k)
{
    if(list.head == nullptr)
    {
        return;
    }
    if(list.head->data == k)
    {
        Node *tmp = list.head;
        list.head = list.head->next;
        delete tmp;
        return;
    }
    Node *tmp1 = list.head;
    Node *tmp2 = tmp1->next;
    while(tmp1 != nullptr && tmp2->data != k)
    {
        tmp1 = tmp1->next;
        tmp2 = tmp1;
    }
    if(tmp2 != nullptr)
    {
        tmp1->next = tmp2->next;
        delete tmp2;
    }
}
void printList1(List list)
{
    while(list.head != nullptr)
    {
        cout << list.head->data << " ";
        list.head = list.head->next;
    }
}
void printList2(Node *tmp)
{
    if(tmp != nullptr)
    {
```



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 22 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

linked list các thứ huhuhuh (cont)

```
> cout << tmp->data << " ";
    printList2(tmp->next);
}
}
//trivial
#include "3_2.h"
node *createANode(int x)
{
    node *NewNode = new node;
    NewNode->data = x;
    NewNode->next = nullptr;
    return NewNode;
}
void loadFile(ifstream &input, LinkedList &list)
{
    list.head = nullptr;
    int data;
    input >> data;
    if (data == 0)
    {
        return;
    }
    node *NewNode = createANode(data);
    list.head = NewNode;
    node *tail = list.head;
    while (true)
    {
        input >> data;
        if (data == 0)
        {
            break;
        }
        NewNode = createANode(data);
        tail->next = NewNode;
        tail = tail->next;
    }
}
```

sdfsadgasdf

djsfgasldkjfhksdhf;sidh;klds



By **emilyle1104**

cheatography.com/emilyle1104/

Not published yet.

Last updated 7th June, 2024.

Page 24 of 24.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

