

Nội dung tuần 6:  
Họ và tên: Nguyễn Huy Đạt  
MSSV: 2305CT2334

Viết chương trình dùng danh sách nối đơn: (Slide 21-31, bài 4, Tài liệu tham khảo chương 18: Linked Lists)

1) Tạo danh sách nodes (nút) có các giá trị thứ tự sau:

13 11 9 7 5 3 1

2) Thêm 1 node (nút) vào đầu danh sách

2 13 11 9 7 5 3 1

3) Thêm 1 node (nút) vào cuối danh sách

2 13 11 9 7 5 3 1 8

4) Hiển thị giá trị của node (nút) ở giữa.

5) Xoá node (nút) thứ 4 (node 3)

## Tuần 6

- Họ và tên: Nguyễn Huy Đạt

- MSSS: 2305CT2334

### Bài tập

```
#include <iostream>
using namespace std;
```

```
struct Node
{
```

```
    int data;
```

```
    Node* next;
```

```
    Node (int value)
```

```
    {
```

```
        data = value;
```

```
        next = nullptr;
```

```
    }
```

```
};
```

```
void printList (Node* head)
```

```
{
```

```
    Node* current = head;
```

```
    while (current != nullptr)
```



```

    {
        cout << current -> data << " ";
        current = current -> next;
    }
    cout << endl;
}

```

```

void addToHead (Node*& head, int value)
{
    Node* newNode = new Node (value);
    newNode -> next = head;
    head = newNode;
}

```

```

void addToTail (Node*& head, int value)
{
    Node* newNode = new Node (value);
    if (head == nullptr)
    {
        head = newNode;
    }
    else
    {
        Node* current = head;
        while (current -> next != nullptr)
        {

```



```

        current = current → next;
    }
    current → next = new Node;
}
}

```

```

void deleteNodeAtIndex (Node * & head, int index)
{

```

```

    if (head == nullptr)
        return;

```

```

    if (index == 1)
    {

```

```

        Node * temp = head;
        head = head → next;
        delete temp;
        return;
    }

```

```

    Node * current = head;

```

```

    int count = 1;

```

```

    while (current != nullptr && count < index - 1)
    {

```

```

        current = current → next;
        count++;
    }

```

```

}

```



Ngày                      No.

```
if (current == nullptr || current -> next == nullptr)
{
    cout << "Không thể xóa, vị trí không hợp lệ." << endl;
    return;
}
```

```
Node* temp = current -> next;
current -> next = temp -> next;
delete temp;
}
```

```
int main()
{
```

```
    int GiaTriKhởiTạo[] = {13, 11, 9, 7, 5, 3, 1};
    int Size = sizeof(GiaTriKhởiTạo) / sizeof(GiaTriKhởiTạo[0]);
```

```
    Node* head = nullptr;
```

```
    for (int i = 0; i < Size; i++)
    {
        addToTail(head, GiaTriKhởiTạo[i]);
    }
```

```
// In danh sách ban đầu
```

```
cout << "\n" << endl;
```

```
cout << "Danh sách ban đầu: ";
```



printList(head);

// 2) Thêm 1 node vào đầu danh sách

addToHead(head, 2);

cout << " \* \* \* \* \* " << endl;

cout << " Sau khi thêm node vào đầu: ";

printList(head);

// 3) Thêm 1 node vào cuối danh sách

addToTail(head, 8);

cout << " \* \* \* \* \* " << endl;

cout << " Sau khi thêm node vào cuối: ";

printList(head);

// 4) Tìm giá trị của 2 node giữa

Node\* slow = head;

Node\* fast = head;

Node\* middle1 = nullptr;

Node\* middle2 = nullptr;

if (head != nullptr)

{

while (fast != nullptr && fast->next != nullptr)

{

fast = fast->next->next;

middle1 = slow;



slow = slow → next;  
}

middle = slow;

if (fast == nullptr)  
{

cout << " \* \* \* \* \* " << endl;

cout << "Gia tri cua 2 node giua: " << middle1 → data  
<< " " << middle2 → data << endl;

}

else

{

cout << " \* \* \* \* \* " << endl;

cout << "Gia tri cua 2 node giua: " << middle1 → data  
<< " " << middle2 → data << endl;

}

}

// 5) Xóa node thứ 4

delete NodeAtIndex(head, 4);

cout << " \* \* \* \* \* " << endl;

cout << "Sau khi xóa node thứ 4: ";

printList(head);

cout << "\n" << endl;

```
Node* current = head;  
while (current != nullptr)  
{
```

```
    Node* next = current->next;  
    delete current;  
    current = next;
```

```
}
```

```
head = nullptr;
```

```
return 0;
```

```
}
```