

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 7 tháng 5 2025, 1:40 PM
Kết thúc lúc	Thứ Tư, 7 tháng 5 2025, 1:56 PM
Thời gian thực hiện	15 phút 56 giây

Câu hỏi 1

Đúng

Cho chương trình khởi tạo, trong đó:

- struct node: đại diện cho một node của linked list
- Hàm createLinkedList:
 - + Nhận vào số phần tử (>0) của link list
 - + Xây dựng một linked list với dữ liệu của các node được nhập từ đầu vào chuẩn (stdin), node mới được thêm vào **CUỐI** linked list
 - + Trả về con trỏ đến node đầu tiên của linked list.
- Hàm isEqual so sánh hai linked list:
 - + Nhận vào con trỏ head1 của linked list thứ nhất, con trỏ của một head2 của linked list thứ 2
 - + Hàm trả về true khi và chỉ khi hai linked list có cùng chiều dài và giá trị của node ở vị trí tương ứng luôn bằng nhau. Ngược lại, hàm trả về false.
- Hàm main đọc vào số phần tử của linked list, gọi hàm createLinkedList để khởi tạo linked list, sau đó gọi hàm print để in ra các phần tử của linked list.

Bạn hãy hoàn thành hàm **isEqual**

Đầu vào:

Các giá trị nhập vào từ đầu vào chuẩn (stdin) đều có giá trị trong khoảng (0; 5000)

Write a program where:

- struct node: represents a node in linkedlists
- functions createLinkedList:
 - + Receive the size of a linked list (>0)
 - + create a linked list with value from standard input (stdin) a new node is add to the **END** of the linked list
 - + Return a pointer which points to the first node of the linked list.
- Function isEqual compare two linked list:
 - + Receive the pointer head1 of the 1st linked list , pointer head2 of the 2nd linked list
 - + Return true if and only if 2 the linked lists have indential size, and node value. Otherwise, return false.
- Function main reads the size of the linked list, calls function createLinkedList to Initialize the linked list, then call functdion print to print all the node's values of the linked list.

Complete function **isEqual**

Input:

The value input from standard input (stdin) with value in (0; 5000)

For example:

Test	Input	Result
1	5 1 3 5 7 9 5 1 3 5 7 9	1

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <iostream>
2 using namespace std;
3 struct node
4 {
5     int data;
6     node *next;
7 };
8
9 node *createLinkedList(int n); // The implementation
10
11 bool isEqual(node *head1, node *head2)
12 {
13     while(head1 != nullptr && head2 != nullptr){
14         if(head1->data != head2->data)
15             return false;
16
17         head1 = head1->next;
18         head2 = head2->next;
19     }
20     return head1 == nullptr && head2 == nullptr;
21 }
22
23 int main()
24 {
25     int n = 0;
26     cin >> n;
27     node *head1 = createLinkedList(n);
28     int m = 0;
29     cin >> m;
30     node *head2 = createLinkedList(m);
31     cout << isEqual(head1, head2) << endl;
32     return 0;
33 }
34

```

	Test	Input	Expected	Got	
✓	1	5 1 3 5 7 9 5 1 3 5 7 9	1	1	✓

Passed all tests! ✓

Câu hỏi 2

Đúng

Cho chương trình khởi tạo, trong đó:

- struct node: đại diện cho một node của linked list
- Hàm print: nhận vào con trỏ head của linked list và in ra từng phần tử của linked list
- Hàm createLinkedList:
 - + Nhận vào số phần tử (>0) của link list
 - + Xây dựng một linked list với dữ liệu của các node được nhập từ đầu vào chuẩn (stdin), node mới được thêm vào **CUỐI** linked list
 - + Trả về con trỏ đến node đầu tiên của linked list.
- Hàm countNode:
 - + Nhận đầu vào là con trỏ head của linked list
 - + Trả về số lượng nodes trong linked list
- Hàm main đọc vào số phần tử của linked list, gọi hàm createLinkedList để khởi tạo linked list, sau đó gọi hàm print để in ra các phần tử của linked list.

Bạn hãy hoàn thành hàm **countNode**

Đầu vào:

Số n là số lượng phần tử trong linked list ($0 < n < 5000$)

n số tiếp theo là giá trị của mỗi node trong linked list, giá trị là một số nguyên có giá trị trong khoảng (-5000; 5000)

Đầu ra:

Thỏa yêu cầu bài toán

Write a program where:

- struct node: represents a node in linkedlists
- functions createLinkedList:
 - + Receive the size of a linked list (>0)
 - + create a linked list with value from standard input (stdin) a new node is add to the **END** of the linked list
 - + Return a pointer which points to the first node of the linked list.
- function countNode:
 - + Input is the head pointer of the linked list
 - + return the number of nodes in linked list
- function main reads the size of the linked list, calls createLinkedList to Initialize the linked list, then calls print to print the linked list.

Complete function **countNode**

Input:

n is size of the linked list ($0 < n < 5000$)

n following numbers are values of each node in the linked list, each value is an integer in (-5000; 5000)

Output:

Satisfy the requirements

For example:

Test	Input	Result
1	5 1 3 5 7 5	1 3 5 7 5 5

Answer: (penalty regime: 0 %)

Reset answer



```

1 #include <iostream>
2 using namespace std;
3 struct node
4 {
5     int data;
6     node *next;
7 };
8
9 node *createLinkedList(int n); // The implementat
10
11 int countNode(node* head)
12 {
13     //TODO
14     int count = 0;
15     while(head){
16         count++;
17         head = head->next;
18     }
19     return count;
20 }
21
22 void print(node *head)
23 {
24     while (head != nullptr)
25     {
26         cout << head->data << endl;
27         head = head->next;
28     }
29 }
30
31 int main()
32 {
33     int n = 0;
34     cin >> n;
35     node *head = createLinkedList(n);
36     print(head);
37     cout<<endl;
38     cout<<countNode(head);
39     return 0;
40 }

```

	Test	Input	Expected	Got	
✓	1	5 1 3 5 7 5	1 3 5 7 5 5	1 3 5 7 5 5	✓

Câu hỏi 3

Đúng

Cho chương trình khởi tạo, trong đó:

- struct node: đại diện cho một node của linked list
- Hàm print: nhận vào con trỏ head của linked list và in ra từng phần tử của linked list
- Hàm createLinkedList:
 - + Nhận vào số phần tử (>0) của link list
 - + Xây dựng một linked list với dữ liệu của các node được nhập từ đầu vào chuẩn (stdin), node mới được thêm vào **ĐẦU** linked list
 - + Trả về con trỏ đến node đầu tiên của linked list.
- Hàm main đọc vào số phần tử của linked list, gọi hàm createLinkedList để khởi tạo linked list, sau đó gọi hàm print để in ra các phần tử của linked list.

Bạn hãy hoàn thành hàm createLinkedList

Đầu vào:

Số n là số lượng phần tử trong linked list ($0 < n < 5000$)

n số tiếp theo là giá trị của mỗi node trong linked list, giá trị là một số nguyên có giá trị trong khoảng $(-5000; 5000)$

Đầu ra:

Thỏa yêu cầu bài toán

English version:

Write a program where:

- struct node: represents a node in linkedlists
- functions createLinkedList:
 - + Receive the size of a linked list (>0)
 - + create a linked list with value from standard input (stdin) a new node is add to the **First** position of the linked list
 - + Return a pointer which points to the first node of the linked list.
- Function main reads the size of the linked list, calls function createLinkedList to Initialize the linked list, then call function print to print the linked list.

Complete function createLinkedList

Input:

size n of the linked list ($0 < n < 5000$)

following by n numbers, each number is a value of a node in the linked list, each number is an integer in $(-5000; 5000)$

Output:

Satisfy the requirements

For example:

Test	Input	Result
1	5 1 3 5 7 9	9 7 5 3 1

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <iostream>
2 using namespace std;
3
4 struct node {
5     int data;
6     node *next;
7 };
8
```

```

9 // Hàm tạo danh sách liên kết từ n phần tử nhập v
10 node *createLinkedList(int n) {
11     if (n <= 0) return nullptr;
12
13     int value;
14     cin >> value;
15     node *head = new node{value, nullptr};
16     node *tail = head;
17
18     for (int i = 1; i < n; ++i) {
19         cin >> value;
20         tail->next = new node{value, nullptr};
21         tail = tail->next;
22     }
23
24     return head;
25 }
26
27 // Hàm in danh sách liên kết theo thứ tự ngược (C
28 void print(node *head) {
29     if (head == nullptr) return;
30     print(head->next);
31     cout << head->data << endl;
32 }
33
34 int main() {
35     int n = 0;
36     cin >> n;
37     if (n > 0) {
38         node *head = createLinkedList(n);
39         print(head);
40     } else {
41         cout << "Invalid n" << endl;
42     }
43     return 0;
44 }
45

```

	Test	Input	Expected	Got	
✓	1	5 1 3 5 7 9	9 7 5 3 1	9 7 5 3 1	✓

Passed all tests! ✓

