

<b>Đã bắt đầu vào lúc</b>	Thứ ba, 5 Tháng mười hai 2023, 10:15 PM
<b>Tình trạng</b>	Đã hoàn thành
<b>Hoàn thành vào lúc</b>	Thứ ba, 5 Tháng mười hai 2023, 10:47 PM
<b>Thời gian thực hiện</b>	32 phút 45 giây
<b>Điểm</b>	3,00/3,00
<b>Điểm</b>	<b>10,00</b> của 10,00 ( <b>100%</b> )

**Câu hỏi****1**

Chính xác

Điểm 1,00 của  
1,00

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 identical 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 function 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 is provided implicitly
10
11 bool isEqual(node *head1, node *head2)
12 {
13     // TO DO
14     bool equal=true;
15     node *temp1=head1;
16     node *temp2=head2;
17     int nodeNum1=0, nodeNum2=0;
18     while (temp1)
19     {
20         temp1 =temp1->next;
21         nodeNum1++;
22     }
23     while (temp2)
24     {
25         temp2=temp2->next;
26         nodeNum2++;
27     }
28     if (nodeNum1==nodeNum2)
29     {

```

```
30     for(int i=1; i<=nodeNum1;i++)
31     {
32         if(head1->data==head2->data)
33         {
34             head1=head1->next;
35             head2=head2->next;
36         }
37         else
38         {equal=false;
39         break;
40
41         }
42     }
43 }
44 else equal=false;
45 return equal;
46
47 }
48
49 int main()
50 {
51     int n = 0;
52     cin>> n;
53     node *head1 = createLinkedList(n);
54     int m = 0;
55     cin>> m;
56     node *head2 = createLinkedList(m);
57     cout << isEqual(head1, head2) << endl;
58     return 0;
59 }
60
```

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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

## Câu hỏi 2

Chính xác

Điểm 1,00 của  
1,00

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

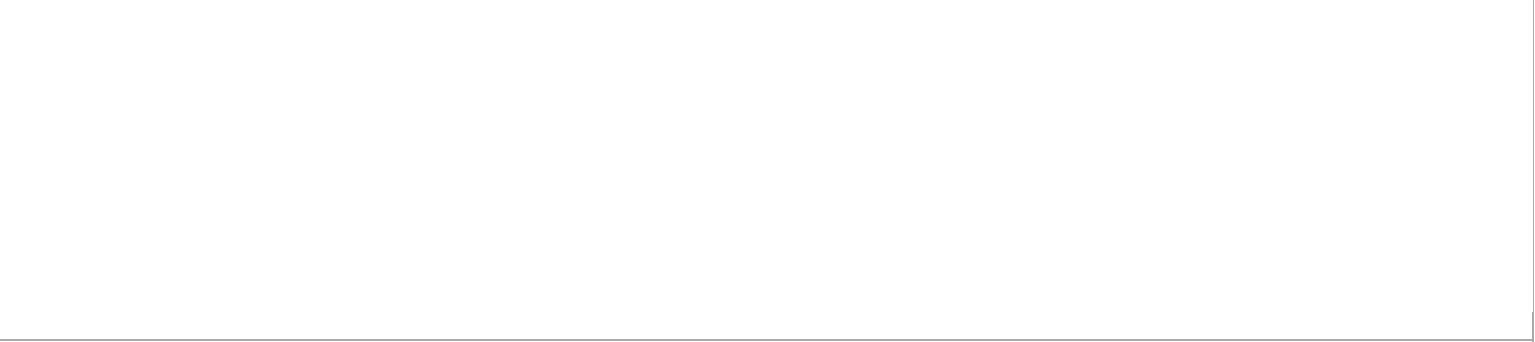
```

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 implementation is provided implicitly
10
11 int countNode(node* head)

```

```
12  {
13      //TODO
14      int nodeNum = 0;
15      while(head)
16      {
17          nodeNum ++;
18          head = head->next;
19      }
20      return nodeNum;
21  }
22  }
23
24  void print(node *head)
25  {
26      while (head != nullptr)
27      {
28          cout << head->data << endl;
29          head = head->next;
30      }
31  }
32  int main()
33  {
34      int n = 0;
35      cin >> n;
36      node *head = createLinkedList(n);
37      print(head);
38      cout<<endl;
39      cout<<countNode(head);
40      return 0;
41  }
42  }
```





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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

## Câu hỏi 3

Chính xác

Điểm 1,00 của  
1,00

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  struct node
4  {
5      int data;
6      node *next;
7  };
8  node *createLinkedList(int n)
9  {
10     // TO DO
11     node* head = nullptr;
12     node* newNode = nullptr;
13
14     for (int i = 0; i < n; i++) {
15         int value;
16         cin >> value;
17
18         newNode = new node;
19         newNode->data = value;
20         newNode->next = head;
21
22         head = newNode;
23     }
24
25     return head;
26

```

```
27 }
28 void print(node *head)
29 {
30     while (head != nullptr)
31     {
32         cout << head->data << endl;
33         head = head->next;
34     }
35 }
36 int main()
37 {
38     int n = 0;
39     cin >> n;
40     if (n > 0)
41     {
42         node *head = createLinkedList(n);
43         print(head);
44     }
45     else
46     {
47         cout << "Invalid n" << endl;
48     }
49     return 0;
50 }
```

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

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

## BÁCH KHOA E-LEARNING



## WEBSITE

HCMUT

MyBK

BKSI

## LIÊN HỆ

📍 268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

☎ (028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

✉ elearning@hcmut.edu.vn

Copyright 2007-2022 BKEL - Phát triển dựa trên Moodle