

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Tư, 7 tháng 5 2025, 3:08 AM
Kết thúc lúc	Thứ Tư, 7 tháng 5 2025, 3:18 AM
Thời gian thực hiện	10 phút 16 giây

Câu hỏi 1

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 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

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 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 n numbers are values of each node in the linked list each number is an integers in $(-5000; 5000)$

Input:

Satisfy the requirements

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```

2 using namespace std;
3 struct node
4 {
5     int data;
6     node *next;
7 };
8 node *createLinkedList(int n)
9 {
10     node *head = nullptr; // Con trỏ đến node đầu tiên
11     node *tail = nullptr; // Con trỏ đến node cuối cùng
12
13     for (int i = 0; i < n; i++)
14     {
15         // Tạo node mới
16         node *newNode = new node;
17
18         // Đọc giá trị cho node mới

```

```

19     cin >> newNode->data;
20
21     // Thiết lập con trỏ next của node mới là nullptr
22     newNode->next = nullptr;
23
24     // Nếu danh sách rỗng, node mới là node đầu tiên
25     if (head == nullptr)
26     {
27         head = newNode;
28         tail = newNode;
29     }
30     // Nếu danh sách không rỗng, thêm node mới vào cuối
31     else
32     {
33         tail->next = newNode;
34         tail = newNode;
35     }
36 }
37
38 return head;
39 }
40 void print(node *head)
41 {
42     while (head != nullptr)
43     {
44         cout << head->data << endl;
45         head = head->next;
46     }
47 }
48 int main()
49 {
50     int n = 0;
51     cin >> n;
52     if (n > 0)
53     {
54         node *head = createLinkedList(n);
55         print(head);
56     }
57     else
58     {
59         cout << "Invalid n" << endl;
60     }
61     return 0;
62 }
63

```

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

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 insertNode
 - + Nhận vào con trỏ head của linked list, con trỏ của một node mới, và vị trí position
 - + Hàm sẽ chèn node mới này vào vị trí position (lấy vị trí của node head là 1). Nếu position <=0, hàm không làm gì cả.

Nếu position lớn hơn số phần tử hiện tại của linked list thì node mới được chèn vào cuối 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 **insertNode**

Đầu vào:

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

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 **END** of the linked list
 - + Return a pointer which points to the first node of the linked list.
- Function insertNode
 - + receive the linked list's head pointers, a pointer of a new node, the position
 - + The function will insert the new node to the input position(head's position is 1). If position <=0, do nothing. If position is greater than the size of the linked list, insert to the end 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 functions **insertNode**

Input:

Value from standards input (stdin) with value in (0; 5000)

For example:

Test	Input	Result
1	5 1 3 5 7 9 2 4	1 3 5 2 7 9

Answer: (penalty regime: 0 %)

Reset answer

```

7  };
8
9  node *createLinkedList(int n); // The implementation is provided implicitly
10
11 node *insertNode(node *head, node *newNode, int position)
12 {
13     // Nếu position <= 0, không làm gì cả và trả về head hiện tại
14     if (position <= 0)
15     {
16         return head;
17     }
18
19     // Nếu position = 1 hoặc danh sách rỗng, chèn vào đầu danh sách

```

```

20     if (position == 1 || head == nullptr)
21     {
22         newNode->next = head;
23         return newNode; // Trả về node mới làm head mới
24     }
25
26     // Trường hợp khác, tìm vị trí để chèn
27     node *current = head;
28     int currentPosition = 1;
29
30     // Tìm node trước vị trí cần chèn hoặc node cuối cùng
31     while (current->next != nullptr && currentPosition < position - 1)
32     {
33         current = current->next;
34         currentPosition++;
35     }
36
37     // Chèn node mới vào sau node current
38     newNode->next = current->next;
39     current->next = newNode;
40
41     // Trả về head ban đầu vì chưa thay đổi
42     return head;
43 }
44
45 void print(node *head)
46 {
47     while (head != nullptr)
48     {
49         cout << head->data << endl;
50         head = head->next;
51     }
52 }
53 int main()
54 {
55     int n = 0;
56     cin >> n;
57     node *head = createLinkedList(n);
58     node *newNode = new node();
59     cin >> newNode->data;
60     int position = 0;
61     cin >> position;
62     head = insertNode(head, newNode, position);
63     print(head);
64     return 0;
65 }
66
67
68

```

	Test	Input	Expected	Got	
✓	1	5 1 3 5 7 9 2 4	1 3 5 2 7 9	1 3 5 2 7 9	✓

Passed all tests! ✓

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 **CUỐI** linked list
 - + Trả về con trỏ đến node đầu tiên của linked list.
- Hàm replace:
 - + Nhận vào con trỏ head của linked list
 - + Vị trí position muốn thay thế (tính từ index 0)
 - + Giá trị value
 - + Hàm replace sẽ thay thế giá trị tại vị trí position bằng giá trị value. Nếu position lớn hơn hoặc bằng số lượng node hoặc position âm thì hàm replace không làm gì cả.
- 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 **replace**

Đầu vào:

Tất cả giá trị từ đầu vào chuẩn (stdin) đều nằm trong khoảng (0, 5000). Riêng giá trị của position có thể âm

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 **END** of the linked list
 - + Return a pointer which points to the first node of the linked list.
- Function replace:
 - + receive the head pointer of the linked list
 - + position that needed to be replaced (count from 0)
 - + new Value
 - + function replace will replace the value at input position to the new Value. If position is equal or greater than the size of the linked list or negative, do nothing.
- Function main reads the linked list's size, calls function createLinkedList to initialize the linked list, then calls function print to print the linked list.

Complete functions **replace**

Input:

All the input from standard input (stdin) with value in (0, 5000). Except position can be negative

Output:

Satisfy the requirements

For example:

Test	Input	Result
1	5 1 3 5 7 9 0 10	1 3 5 7 9 10 3 5 7 9

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 void replace(node* head, int position, int value)
12 {
13     // Nếu vị trí âm hoặc danh sách rỗng, không làm gì cả
14     if (position < 0 || head == nullptr)
15     {
16         return;
17     }
18
19     // Duyệt đến node ở vị trí position
20     node* current = head;
21     int currentPos = 0;
22
23     while (current != nullptr && currentPos < position)
24     {
25         current = current->next;
26         currentPos++;
27     }
28
29     // Nếu đã đến cuối danh sách mà chưa đến vị trí cần thay thế
30     // (nghĩa là position >= số lượng node), không làm gì cả
31     if (current == nullptr)
32     {
33         return;
34     }
35
36     // Thay thế giá trị tại vị trí position
37     current->data = value;
38 }
39
40
41 void print(node *head)
42 {
43     while (head != nullptr)
44     {
45         cout << head->data << " ";
46         head = head->next;
47     }
48     cout<<endl;
49 }
50 int main()
51 {
52     int n = 0;
53     cin >> n;
54     node *head = createLinkedList(n);
55     print(head);
56     int pos, val;
57     cin>>pos>>val;
58     replace(head, pos, val);
59     print(head);
60     return 0;
61 }
62

```

	Test	Input	Expected	Got	
✓	1	5 1 3 5 7 9 0 10	1 3 5 7 9 10 3 5 7 9	1 3 5 7 9 10 3 5 7 9	✓

Passed all tests! ✓

Câu hỏi 4

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 searchLinkedList:
 - + Nhận vào con trỏ head của linked list
 - + Giá trị cần tìm
 - + Nếu tìm thấy thì trả về vị trí đầu tiên của nó trong Linked List (index từ 0), không thì trả về -1.
- 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 **searchLinkedList**

Đầu vào:

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

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 **END** of the linked list
 - + Return a pointer which points to the first node of the linked list.
- function searchLinkedList:
 - + receive the linked list's head pointer
 - + The searching Value
 - + If found, return the first position the value appeared in the Linked List (index start from 0), otherwise -1.
- function main read the linked list's size, calls function createLinkedList to initialize the linked list, then calls function print to print the linked list.

Complete functions **searchLinkedList**

Input:

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

For example:

Test	Input	Result
1	5	1
	1 3 5 7 9	3
	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 int searchLinkedList(node* head, int key)

```



```

12 {
13     // Kiểm tra nếu danh sách rỗng
14     if (head == nullptr)
15     {
16         return -1;
17     }
18
19     // Duyệt qua từng node trong linked list
20     node* current = head;
21     int position = 0;
22
23     while (current != nullptr)
24     {
25         // Nếu tìm thấy giá trị key, trả về vị trí hiện tại
26         if (current->data == key)
27         {
28             return position;
29         }
30
31         // Chuyển đến node tiếp theo và tăng vị trí
32         current = current->next;
33         position++;
34     }
35
36     // Nếu không tìm thấy key trong linked list
37     return -1;
38 }
39
40 void print(node *head)
41 {
42     while (head != nullptr)
43     {
44         cout << head->data << endl;
45         head = head->next;
46     }
47 }
48 int main()
49 {
50     int n = 0;
51     cin >> n;
52     node *head = createLinkedList(n);
53     print(head);
54     int m;
55     cin>>m;
56     cout<<searchLinkedList(head,m);
57     return 0;
58 }
59

```



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

Passed all tests! ✓