

|                     |                                   |
|---------------------|-----------------------------------|
| Trạng thái          | Đã xong                           |
| Bắt đầu vào lúc     | Thứ Năm, 8 tháng 5 2025, 3:27 PM  |
| Kết thúc lúc        | Thứ Bảy, 10 tháng 5 2025, 7:40 AM |
| Thời gian thực hiện | 1 ngày 16 giờ                     |

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<br>1 3 5 7 9 | 1<br>3<br>5<br>7<br>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 node *createLinkedList(int n)
9 {
10     // TO DO
11     node *head = nullptr;
12     node *tail = nullptr;
13
14     for(int i = 0; i < n; ++i){
15         int value;
16         cin >> value;
17
18         node *newNode = new node;
19         newNode->data = value;
20         newNode->next = nullptr;
21
22         if(head == nullptr){
23             head = tail = newNode;
24         }else{
25             tail->next = newNode;
26             tail = newNode;
27         }
28     }
29     return head;
30 }
31 void print(node *head)
32 {
33     while (head != nullptr)
34     {
35         cout << head->data << endl;
36         head = head->next;
37     }
38 }
39 int main()
40 {
41     int n = 0;
42     cin >> n;
43     if (n > 0)
44     {
45         node *head = createLinkedList(n);
46         print(head);
47     }
48     else
49     {
50         cout << "Invalid n" << endl;
51     }
52     return 0;
53 }
54

```

|   | Test | Input          | Expected              | Got                   |   |
|---|------|----------------|-----------------------|-----------------------|---|
| ✓ | 1    | 5<br>1 3 5 7 9 | 1<br>3<br>5<br>7<br>9 | 1<br>3<br>5<br>7<br>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 nodei, 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<br>1 3 5 7 9<br>2 4 | 1<br>3<br>5<br>2<br>7<br>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 implement
10
11 node *insertNode(node *head, node *newNode, int position)
12 {
13     // TO DO
14     if (position <= 0) {
15         return head;
16     }
17
18     // Chèn vào đầu danh sách
19     if (position == 1) {
20         newNode->next = head;
21         head = newNode;
22         return head;
23     }
24
25     node *temp = head;
26     int currentPos = 1;
27
28     // Duyệt qua danh sách đến vị trí cần chèn
29     while (temp != nullptr && currentPos < position) {
30         temp = temp->next;
31         currentPos++;
32     }
33
34     // Nếu vị trí hợp lệ, chèn node vào danh sách
35     if (temp != nullptr) {
36         newNode->next = temp->next;
37         temp->next = newNode;
38     } else {
39         // Nếu position lớn hơn số phần tử, thêm
40         temp = head;
41         while (temp->next != nullptr) {
42             temp = temp->next;
43         }
44         temp->next = newNode;
45     }
46
47     return head;
48 }
49 void print(node *head)
50 {
51     while (head != nullptr)
52     {
53         cout << head->data << endl;
54         head = head->next;
55     }
56 }
57 int main()
58 {
59     int n = 0;
60     cin >> n;
61     node *head = createLinkedList(n);
62

```

|   | Test | Input                 | Expected                   | Got                        |   |
|---|------|-----------------------|----------------------------|----------------------------|---|
| ✓ | 1    | 5<br>1 3 5 7 9<br>2 4 | 1<br>3<br>5<br>2<br>7<br>9 | 1<br>3<br>5<br>2<br>7<br>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<br>1 3 5 7 9<br>0 10 | 1 3 5 7 9<br>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 implementat
10
11 void replace(node* head, int position, int value)
12 {
13     //TODO
14     if (position < 0) return; // Nếu position âm,
15
16     node *temp = head;
17     int currentPos = 0;
18
19     // Duyệt qua danh sách đến vị trí cần thay th
20     while (temp != nullptr && currentPos < positi
21         temp = temp->next;
22         currentPos++;
23     }
24
25     // Nếu vị trí hợp lệ, thay đổi giá trị của n
26     if (temp != nullptr) {
27         temp->data = value;
28     }
29 }
30 void print(node *head)
31 {
32     while (head != nullptr)
33     {
34         cout << head->data << " ";
35         head = head->next;
36     }
37     cout<<endl;
38 }
39 int main()
40 {
41     int n = 0;
42     cin >> n;
43     node *head = createLinkedList(n);
44     print(head);
45     int pos, val;
46     cin>>pos>>val;
47     replace(head, pos, val);
48     print(head);
49     return 0;
50 }
51
```

|   | Test | Input                  | Expected                | Got                     |   |
|---|------|------------------------|-------------------------|-------------------------|---|
| ✓ | 1    | 5<br>1 3 5 7 9<br>0 10 | 1 3 5 7 9<br>10 3 5 7 9 | 1 3 5 7 9<br>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<br>1 3 5 7 9<br>3 | 1<br>3<br>5<br>7<br>9<br>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 implementat
10
11 int searchLinkedList(node* head, int key)
12 {
13     // TODO
14     int index = 0;
15     node *current = head;
16
17     // Duyệt qua từng node và kiểm tra giá trị
18     while (current != nullptr) {
19         if (current->data == key) {
20             return index; // Nếu tìm thấy, trả v
21         }
22         current = current->next;
23         index++;
24     }
25
26     return -1; // Nếu không tìm thấy, trả về -1
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     node *head = createLinkedList(n);
41     print(head);
42     int m;
43     cin>>m;
44     cout<<searchLinkedList(head,m);
45     return 0;
46 }
47

```

|   | Test | Input               | Expected                   | Got                        |   |
|---|------|---------------------|----------------------------|----------------------------|---|
| ✓ | 1    | 5<br>1 3 5 7 9<br>3 | 1<br>3<br>5<br>7<br>9<br>1 | 1<br>3<br>5<br>7<br>9<br>1 | ✓ |

Passed all tests! ✓



