Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Sáu, 16 tháng 5 2025, 10:41 PM
Kết thúc lúc	Thứ Sáu, 16 tháng 5 2025, 11:53 PM
Thời gian thực hiện	1 giờ 12 phút
Điểm	30,00/30,00
Điểm	**10,00** trên 10,00 (**100**%)

Câu hỏi 1

Đúng

Đạt điểm 10,00 trên 10,00

Một dãy số nguyên được cài đặt bằng danh sách liên kết như sau:

```
struct Node
{
    int value;
    Node* next;
};
```

Biết $head$ là con trỏ trỏ tới $node$ đầu tiên của danh sách liên kết, viết hàm`Node* removeMod(Node* head, int m)` để xóa tất cả các node mà $value$ của nó chia hết cho giá trị m. Hàm trả về con trỏ trỏ tới đầu danh sách liên kết. Nếu xóa hết tất cả các node, trả về giá trị NULL. m không thể có giá trị 0.

For example:

Input	Result
5 1 2 3 4 5 2	1 3 5
5 1 7 5 8 12 1	

Answer: (penalty regime: 0 %)

Reset answer

```
Node *removeMod(Node *head, int m) {
  Node* newhead = new Node;
  Node* p = newhead;
  while (head) {
    if (head->value % m) {
      p->next = new Node;
      p->next->value = head->value;
      p = p->next;
    }
    Node *x = head;
    head = head->next;
    delete x;
  }
  Node *x = newhead;
  newhead = newhead->next;
  delete x;
  return newhead;
}
```

	Input	Expected	Got	
✓	5 1 2 3 4 5 2	1 3 5	1 3 5	✓

	Input	Expected	Got	
✓	5 1 7 5 8 12 1			✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 2

Đúng

Đạt điểm 10,00 trên 10,00

Một dãy số nguyên được cài đặt bằng danh sách liên kết như sau:

```
struct Node
{
    int value;
    Node* next;
};
```

Biết $head$ là con trỏ trỏ tới $node$ đầu tiên của danh sách liên kết, viết hàm **Node* swapAfter(Node* head, int x)** để đổi chỗ những node có giá trị x với node đằng sau nó. Nếu không có node có giá trị x hoặc là node cuối thì không làm gì cả. Hàm trả về con trỏ trỏ tới đầu danh sách liên kết.

Bổ sung: Nếu node có giá trị x và node liền sau nó cũng có giá trị x thì không làm gì cả.

Lưu ý : bạn phải viết code đổi chỗ 2 node chứ không phải đổi giá trị 2 node cho nhau.

For example:

Input	Result
5 1 2 3 4 5 3	1 2 4 3 5
5 1 2 4 1 3 1	2 1 4 3 1

Answer: (penalty regime: 0 %)

Reset answer

```
Node *swapAfter(Node *head, int x) {
  Node* dummy = new Node;
  dummy->next = head;
  Node* prev = dummy;
  while (prev->next && prev->next->next) {
    Node *cur = prev->next;
    Node *nxt = cur->next;
    if (cur->value == x && nxt->value != x) {
      cur->next = nxt->next;
      nxt->next = cur;
      prev->next = nxt;
      prev = cur;
    } else {
      prev = prev->next;
    }
  }
  return dummy->next;
}
```

	Input	Expected	Got	
✓	5 1 2 3 4 5 3	1 2 4 3 5	1 2 4 3 5	✓
✓	5 1 2 4 1 3 1	2 1 4 3 1	2 1 4 3 1	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 3

Đúng

Đạt điểm 10,00 trên 10,00

Cho một `struct Node` biểu diễn một $node$ của 1 danh sách liên kết đơn như sau:

```
struct Node
{
    int value;
    Node* next;
};
```

Biết $head$ là con trỏ trỏ tới $node$ đầu tiên của danh sách liên kết, viết hàm:

- `Node* splitEvenOdd(Node* head);` sửa đổi danh sách được liên kết sao cho tất cả các số chẵn xuất hiện trước tất cả các số lẻ trong danh sách được liên kết đã sửa đổi. Ngoài ra, việc sửa đổi phải giữ nguyên thứ tự của số chẵn và số lẻ trong danh sách. Hàm trả về con trỏ trỏ tới đầu danh sách liên kết.

Ví dụ:

- **Input**: 3->2->8->5->7->NULL
- **Output**: 2->8->3->5->7->NULL

Lưu ý : bạn phải viết code thay đổi liên kết giữa các node chứ không phải thay đổi giá trị giữa các node.

For example:

Test	Input	Result
`head = splitEvenOdd(head);` `// printList(head);`	6 470 372 405 111 829 474	470 372 474 405 111 829

Answer: (penalty regime: 0 %)

Reset answer

```
Node* splitEvenOdd(Node* head) {
  Node *newhead = new Node;
  Node *p = newhead;
  Node *q = head;
  while (q) {
    if (q->value % 2 == 0) {
      p->next = new Node;
      p->next->value = q->value;
      p = p->next;
    }
    q = q->next;
  }
  while (head) {
    if (head->value % 2) {
      p->next = new Node;
      p->next->value = head->value;
      p = p->next;
    }
    Node *x = head;
    head = head->next;
    delete x;
  }
  Node *x = newhead;
  newhead = newhead->next;
  delete x;
  return newhead;
}

```

	Test	Input	Expected	Got	
✓	head = splitEvenOdd(head); // printList(head);	6 470 372 405 111 829 474	470 372 474 405 111 829	470 372 474 405 111 829	✓

✓	head = splitEvenOdd(head);
	head = insertHead(head, 77);
	head = splitEvenOdd(head);
		3
		286 436 775
			286 436 77 775	286 436 77 775	✓

✓	head = splitEvenOdd(head);	3 13 19 20	20 13 19	20 13 19	✓
✓	head = splitEvenOdd(head);	9 2 4 6 8 10 1 3 5 7	2 4 6 8 10 1 3 5 7	2 4 6 8 10 1 3 5 7	✓
✓	head = splitEvenOdd(head);	10 1 3 5 7 9 2 4 6 8 10	2 4 6 8 10 1 3 5 7 9	2 4 6 8 10 1 3 5 7 9	✓
✓	head = splitEvenOdd(head);	5 2 4 6 8 10	2 4 6 8 10	2 4 6 8 10	✓

Passed all tests! ✓

▾ SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)

```cpp
Node* splitEvenOdd(Node *head) {
    if(!head) return nullptr;

    Node *p = head;
    Node *last_even = nullptr;

    while (p)
    {
        if(p -> value % 2== 0) last_even = p;
        p = p -> next;
    }
    ///1 3 5 7 9
    if(!last_even) return head;///chi gom cac so le

    p = last_even;
    while (head->value %2 == 1 && head != last_even)///1 3 5 7 8
    {
        Node *tmp = head;
        head = head->next;
        tmp ->next = p->next;
        p->next = tmp;
        p = p->next;
    }

    if(head != last_even)
    {
        Node *q = head;
        while (q != last_even)///1 3 5 7 8
        {
            bool flag = false;
            if(q->next ->value %2 == 1){
                flag = true;
                Node *tmp = q->next;
                q->next = q->next->next;
                tmp ->next = p->next;
                p->next = tmp;
                p = p->next;
            }
           if(!flag) q = q->next;
        }
    }

    return head;
}
```

Đúng

Marks for this submission: 10,00/10,00.

Trở lại Khoá học