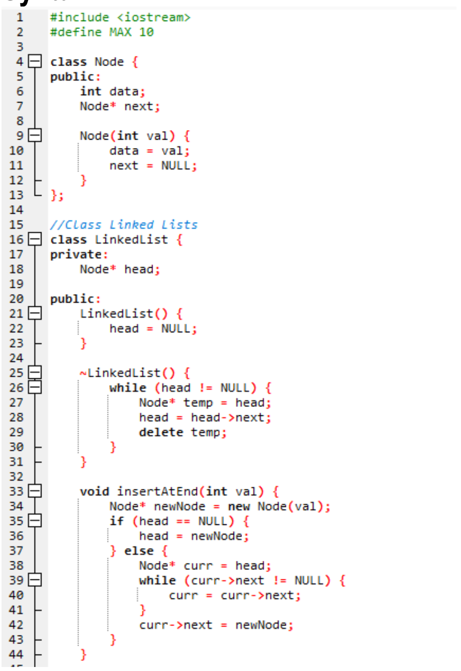
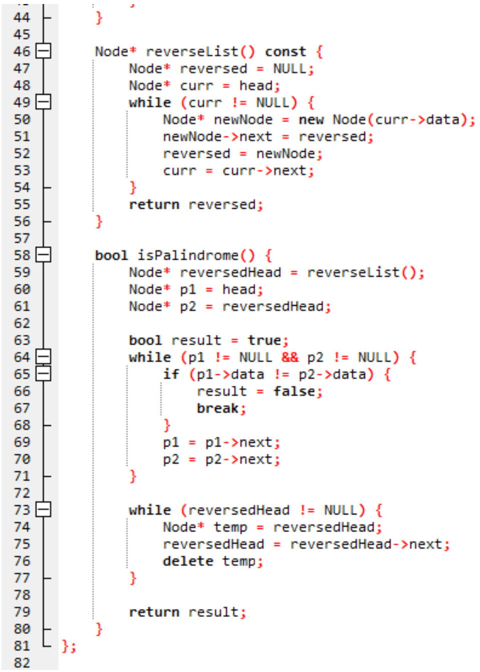
Activity No. <n>	
<Prelim Laboratory Exam>	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed: 9/2/25**
Section: CPE21S4	**Date Submitted: 9/2/25**
Name(s): Alcantara, Jason P.	**Instructor: Engr. Jimlord Quejado**

6. Output:

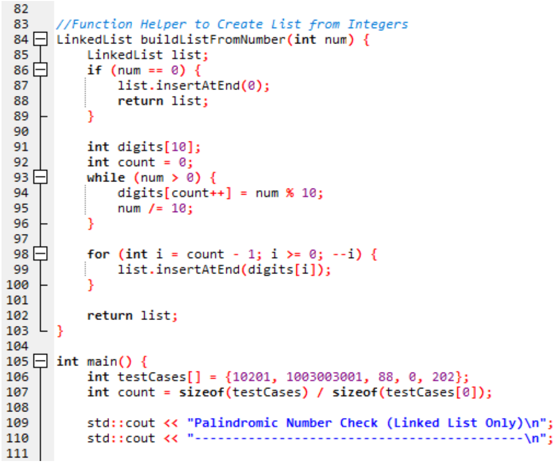
Syntax:



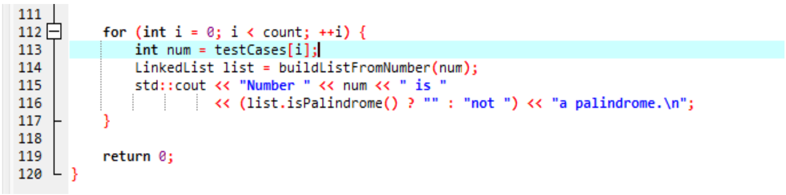
```cpp
#include <iostream>
#define MAX 10

class Node {
public:
    int data;
    Node* next;

    Node(int val) {
        data = val;
        next = NULL;
    }
};

//Class Linked Lists
class LinkedList {
private:
    Node* head;

public:
    LinkedList() {
        head = NULL;
    }

    ~LinkedList() {
        while (head != NULL) {
            Node* temp = head;
            head = head->next;
            delete temp;
        }
    }

    void insertAtEnd(int val) {
        Node* newNode = new Node(val);
        if (head == NULL) {
            head = newNode;
        } else {
            Node* curr = head;
            while (curr->next != NULL) {
                curr = curr->next;
            }
            curr->next = newNode;
        }
    }
```



```cpp
    }

    Node* reverseList() const {
        Node* reversed = NULL;
        Node* curr = head;
        while (curr != NULL) {
            Node* newNode = new Node(curr->data);
            newNode->next = reversed;
            reversed = newNode;
            curr = curr->next;
        }
        return reversed;
    }

    bool isPalindrome() {
        Node* reversedHead = reverseList();
        Node* p1 = head;
        Node* p2 = reversedHead;

        bool result = true;
        while (p1 != NULL && p2 != NULL) {
            if (p1->data != p2->data) {
                result = false;
                break;
            }
            p1 = p1->next;
            p2 = p2->next;
        }

        while (reversedHead != NULL) {
            Node* temp = reversedHead;
            reversedHead = reversedHead->next;
            delete temp;
        }

        return result;
    }
};

```

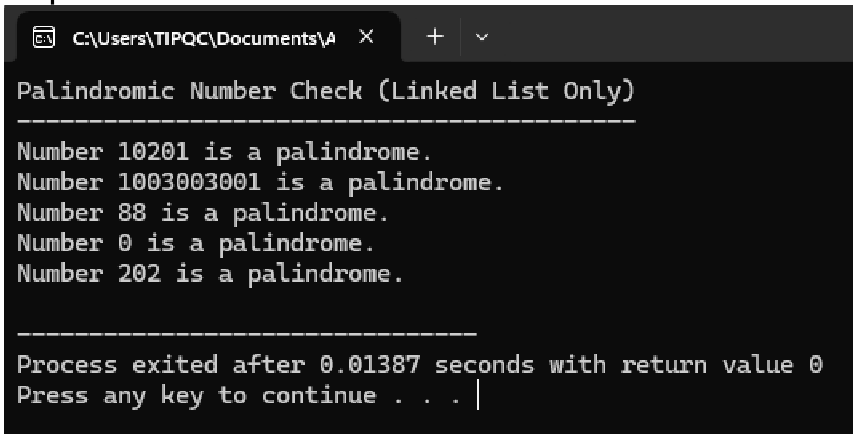


```cpp

//Function Helper to Create List from Integers
LinkedList buildListFromNumber(int num) {
    LinkedList list;
    if (num == 0) {
        list.insertAtEnd(0);
        return list;
    }

    int digits[10];
    int count = 0;
    while (num > 0) {
        digits[count++] = num % 10;
        num /= 10;
    }

    for (int i = count - 1; i >= 0; --i) {
        list.insertAtEnd(digits[i]);
    }

    return list;
}

int main() {
    int testCases[] = {10201, 1003003001, 88, 0, 202};
    int count = sizeof(testCases) / sizeof(testCases[0]);

    std::cout << "Palindromic Number Check (Linked List Only)\n";
    std::cout << "-------------------------------------------\n";

```

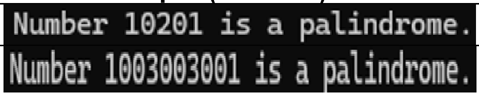
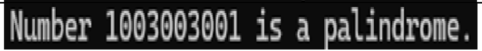
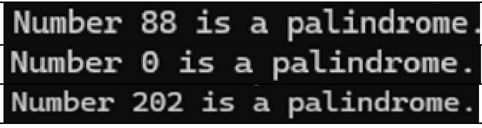
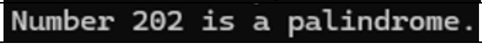


```cpp

    for (int i = 0; i < count; ++i) {
        int num = testCases[i];
        LinkedList list = buildListFromNumber(num);
        std::cout << "Number " << num << " is "
                  << (list.isPalindrome() ? "" : "not ") << "a palindrome.\n";
    }

    return 0;
}
```

Output:



```
Palindromic Number Check (Linked List Only)
-------------------------------------------
Number 10201 is a palindrome.
Number 1003003001 is a palindrome.
Number 88 is a palindrome.
Number 0 is a palindrome.
Number 202 is a palindrome.

--------------------------------
Process exited after 0.01387 seconds with return value 0
Press any key to continue . . .
```

Palindrome?	Output (Console)	Remark
10201	 Number 10201 is a palindrome.	Yes it is palindrome
1003003001	 Number 1003003001 is a palindrome.	Yes it is palindrome
88	Number 88 is a palindrome.	Yes it is palindrome
0	 Number 0 is a palindrome.	Yes it is palindrome
202	 Number 202 is a palindrome.	Yes it is palindrome

7. Supplementary Activity

8. Conclusion:

In This preliminary exam we make create a code about palindrome using Array- Based or Linked List this exam I get confused because I need to re-read the whole instruction to make the code but eventually I finished the task using linked list and this examination I get challenged because I need to go back to the topic we discussed so that I can finish It.

9. Assessment Rubric