Find the factorial of Number Using a linked list

Intro

Factorial of a number (n!) is the product of all positive integer from 1 to n
For a large number factorial values grow exponentially exceeding the storage limit of standard data types

To handle large factorial values we use a linked list each node stores. of the number.

Why Use a linked list ?

→ 1. Factorial Values become too large for int or long long
→ A linked list dynamically memory avoiding Overflow.
→ we can store each digit seperately in a node making calculations effecients.

Approach.

1. Use a linked list to store the factorial in reverse order
2. Iterate from 1 to n and multiply the number using list traversal
3. traverse the linked list to get the final factorial value.

implementation Steps.

① Define a Node structure
→ each node contains a digits of the factorial

② Multiply the current number with the linked list representation of factorial
→ Traverse the linked list & update value
→ Maintain a carry for overflow values

③ Reverse the linked to display the factorial correctly

```
#include <iostream>
using namespace std;

Struct Node {
    int data;
    Node* next;
};

Node* insertHead (Node* head, int value){
    Node* newNode = new Node(value, Nu
    if(!head) return newNode;
    Node* temp = head;
    while (temp->next) temp = temp->next
    temp->next = NewNode;
    return head;
}

Node* multiply (Node* head, int num
    Node* temp = head, *prev = NULL;
```

```
int carry=0;
while(temp){
    int prod = temp->data * num + carry;
    temp->data = prod % 10;
    carry = prod / 10;
    prev = temp;
    temp = temp->next;
}
while(carry){
    prev->next = new Node{carry%10, NULL};
    carry /= 10;
    prev = prev->next;
}
return head;
}

Node* reverseList(Node* head){
    Node* prev = NULL, *curr = head, *next =
        NULL;
    while(curr){
        next = curr->next
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    return prev;
}

void factorial(int n){
    Node* head = new Node{1, NULL};

    for(int i = 2; i <= n; i++){
```

```
        head = multiply(head, i);
    }
    head = reverselist(head);

    while (head){
        cout << head->data;
        head = head->next;
    }
    }
    cout << endl;
}

int main() {
    int n;
    cout << "Enter a number: ";
    Cin >> n;
    cout << "Factorial of " << n << " is: ";
    factorial(n);
    return 0;
}
```

Explaination of code.

→ insertEnd() - Add a node at the end of the linked list

→ multiply() - Multiplies the number stored in the linked list by a given integer

→ reverseList() - Reverse the linked list to display the number correctly

→ factorial(n) - Computes factorial using linked list multiplication

main() → Takes input from the user & Calls the factorial() function

Complexity Analysis.
Multiplication Complexity O(n×m) where n is the number and m is the number of digit
Memory : O(m) where m is the no of digits in the factorial

Advantages of Using a linked list for factorial
→ Handle large number effeciently
→ Memory effecient since it only stores necessary digit
→ prevents integer overflow