DAY 3

Maadhav Hira **22BCS10380** **KPIT-901(A)**

Q1 . Add Two Numbers

day3ques1.cpp > reverseList(ListNode *)

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

struct ListNode {
    int val;
    ListNode* next;
    ListNode() : val(0), next(nullptr) {}
    ListNode(int x) : val(x), next(nullptr) {}
    ListNode(int x, ListNode* next) : val(x), next(next) {}
};

ListNode* reverseList(ListNode* head) {
    ListNode* prev = nullptr;
    ListNode* curr = head;

    while (curr) {
        ListNode* next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    return prev;
}

long long listToNumber(ListNode* head) {
    long long num = 0;
    while (head) {
        num = num * 10 + head->val;
        head = head->next;
    }
    return num;
}

int main() {
    ListNode* l1 = new ListNode(2, new ListNode(4, new ListNode(3)));
    ListNode* l2 = new ListNode(5, new ListNode(6, new ListNode(4)));
    l1 = reverseList(l1);
    l2 = reverseList(l2);

    long long num1 = listToNumber(l1);
    long long num2 = listToNumber(l2);
    long long sum = num1 + num2;
    cout << "Sum: " << sum << endl;

    return 0;
}
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day3ques1.cpp -o day3ques1 && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\"day3ques1
Sum: 807
```

Q2. Reverse Linked List

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

struct ListNode {
    int val;
    ListNode* next;
    ListNode(int x) : val(x), next(nullptr) {}
};

ListNode* reverseList(ListNode* head) {
    ListNode* prev = nullptr;
    ListNode* current = head;

    while (current != nullptr) {
        ListNode* nextNode = current->next;
        current->next = prev;
        prev = current;
        current = nextNode;
    }

    return prev;
}

void printList(ListNode* head) {
    while (head != nullptr) {
        cout << head->val << " ";
        head = head->next;
    }
    cout << endl;
}
ListNode* createList(int n) {
    if (n == 0) return nullptr;
    int val;
    cout << "Enter the value of node 1: ";
    cin >> val;
    ListNode* head = new ListNode(val);
    ListNode* current = head;
    for (int i = 2; i <= n; ++i) {
        cout << "Enter the value of node " << i << ": ";
        cin >> val;
        current->next = new ListNode(val);
        current = current->next;
    }
    return head;
}
```

```cpp
int main() {
    int n;
    cout << "Enter the number of nodes in the linked list: ";
    cin >> n;

    ListNode* head = createList(n);

    cout << "Original Linked List: ";
    printList(head);

    head = reverseList(head);

    cout << "Reversed Linked List: ";
    printList(head);

    return 0;
}
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day3ques2.cpp -o day3ques2 && "c:\Users\Gaurav Kumar\OneDr
ive\Desktop\class\"day3ques2
Enter the number of nodes in the linked list: 5
Enter the value of node 1: 2
Enter the value of node 2: 5
Enter the value of node 3: 6
Enter the value of node 4: 7
Enter the value of node 5: 5
Original Linked List: 2 5 6 7 5
Reversed Linked List: 5 7 6 5 2
```

Q3. Fibonacci Series Using Recursion

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

int fibonacci(int n) {
    if (n <= 1) return n;
    return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
    int n;
    cout << "Enter the value of n: ";
    cin >> n;
    cout << "Fibonacci number F(" << n << ") = " << fibonacci(n) << endl;
    return 0;
}
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day3ques3.cpp -o day3ques3 && "c:\Users\Gaurav Kumar\OneDr
ive\Desktop\class\"day3ques3
Enter the value of n: 8
Fibonacci number F(8) = 21
```

Q4.Power Of Three

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

bool isPowerOfThree(int n) {
    if (n <= 0) return false;
    while (n % 3 == 0) {
        n /= 3;
    }
    return n == 1;
}

int main() {
    int n;
    cout << "Enter an integer: ";
    cin >> n;
    if (isPowerOfThree(n)) {
        cout << n << " is a power of three." << endl;
    } else {
        cout << n << " is not a power of three." << endl;
    }
    return 0;
}
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day3ques4.cpp -o day3ques4 && "c:\Users\Gaurav Kumar\OneDr
ive\Desktop\class\"day3ques4
Enter an integer: 27
27 is a power of three.
```

Q5. Wildcard Matching

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

bool isMatch(string s, string p) {
    int m = s.size(), n = p.size();

    vector<vector<bool>> dp(m + 1, vector<bool>(n + 1, false));

    dp[0][0] = true;

    for (int j = 1; j <= n; ++j) {
        if (p[j - 1] == '*') {
            dp[0][j] = dp[0][j - 1];
        }
    }

    for (int i = 1; i <= m; ++i) {
        for (int j = 1; j <= n; ++j) {
            if (p[j - 1] == '*') {
                dp[i][j] = dp[i - 1][j] || dp[i][j - 1];
            } else if (p[j - 1] == '?' || s[i - 1] == p[j - 1]) {
                dp[i][j] = dp[i - 1][j - 1];
            }
        }
    }

    return dp[m][n];
}

int main() {
    string s, p;
    cout << "Enter the string: ";
    cin >> s;
    cout << "Enter the pattern: ";
    cin >> p;

    if (isMatch(s, p)) {
        cout << "Pattern matches the string." << endl;
    } else {
        cout << "Pattern does not match the string." << endl;
    }

    return 0;
}

```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day3ques5.cpp -o day3ques5 && "c:\Users\Gaurav Kumar\OneDr
ive\Desktop\class\"day3ques5
Enter the string: asd
Enter the pattern: asd
Pattern matches the string.
```