DAY 3

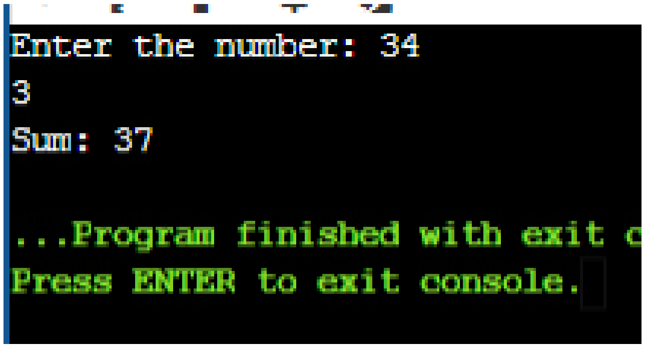
Student Name: Suman Kumar UID: 22BCS15488
Branch: BE-CSE Section/Group: 620 - B
Date of Performance:19/12/24

Problem 1

1. Aim: Sum of two numbers
2. Problem Statement: Given a number n, find sum of first n natural numbers. To calculate the sum, we will use a recursive function recur_sum().
3. Task: Given an integer n, print the sum of two numbers.
4. Code:

```
#include<iostream>
using namespace std;
int sum(int i , int j)
{
      return i+j;
}
int main()
{
     Int x,y;
cout<<“Enter the numbers: ”;
cin>>x>>y;
cout<<“Sum: ”<<sum(x,y);
return 0;
}
```

5. Output:



```
Enter the number: 34
3
Sum: 37

...Program finished with exit c
Press ENTER to exit console.
```

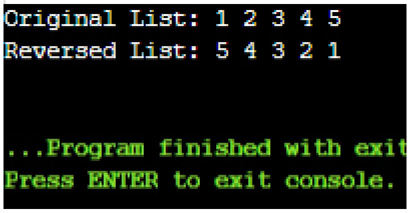
Problem 2

1. Aim: Reverse the linked list and return the reversed list

2. Task: Given the head of a singly linked list, reverse the list, and return the reversed list.
3. Code:

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

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

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

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

    return prev;
}

void printList(ListNode* head) {
    while (head != nullptr) {
        cout << head->val << " ";
        head = head->next;
    }
    cout << endl;
}

int main() {
    ListNode* head = new ListNode(1);
    head->next = new ListNode(2);
    head->next->next = new ListNode(3);
    head->next->next->next = new ListNode(4);
    head->next->next->next->next = new ListNode(5);

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

    head = reverseList(head);

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

    return 0;
}
```

5. Output:



```
Original List: 1 2 3 4 5
Reversed List: 5 4 3 2 1


...Program finished with exit
Press ENTER to exit console.
```

Problem 3

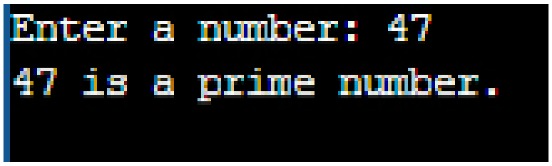
1. Aim: Write a function to check number is prime or not

2. Code:

```
#include<iostream>

using namespace std;
```

```
bool isPrime(int num) {
    if (num <= 1) return false;
    for (int i = 2; i * i <= num; ++i) {
        if (num % i == 0) return false;
    }
    return true;
}
int main() {
    int number;
    cout<<"Enter a number: ";
    cin>>number;
    if (isPrime(number)) {
        cout << number << " is a prime number." << endl;
    } else {
        cout << number << " is not a prime number." << endl;
    }
    return 0;
}
```

3. Output:



```
Enter a number: 47
47 is a prime number.
```

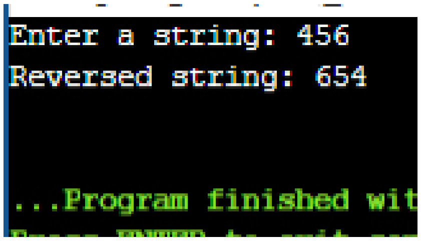
Problem 4

1. Aim: Write a function to reverse the string.

2. Code:

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

string reverseString(string str) {
  string reversedStr = "";
  for (int i = str.length() - 1; i >= 0; i--) {
    reversedStr += str[i];
  }
  return reversedStr;
}

int main() {
  string inputString;

  cout << "Enter a string: ";
  getline(cin, inputString);

  string reversedString = reverseString(inputString);

  cout << "Reversed string: " << reversedString << endl;

  return 0;
}
```

5. Output:



```
Enter a string: 456
Reversed string: 654


...Program finished wit
```

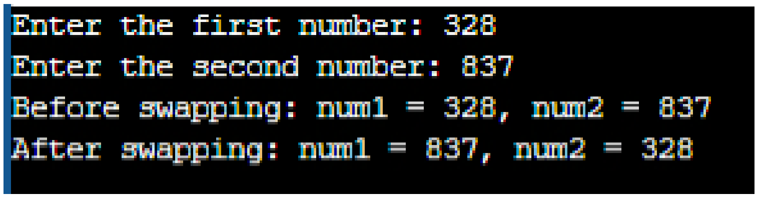
Problem 5

1. Aim: To implement the function to swap two variables using pass by reference.

2. Code:

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

void swap(int& a, int& b) {
  int temp = a;
  a = b;
  b = temp;
}
int main() {
  int num1, num2;

  cout << "Enter the first number: ";
  cin >> num1;
  cout << "Enter the second number: ";
  cin >> num2;
  cout << "Before swapping: num1 = " << num1 << ", num2 = " << num2 << endl;
  swap(num1, num2);
  cout << "After swapping: num1 = " << num1 << ", num2 = " << num2 << endl;

  return 0;
}
```

5. Output:

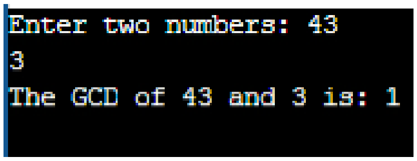


```
Enter the first number: 328
Enter the second number: 837
Before swapping: num1 = 328, num2 = 837
After swapping: num1 = 837, num2 = 328
```

Problem 6

1. Aim: Write a recursive function to find the GCD of two numbers
2. Code:

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

int gcd(int a, int b) {
  while (b != 0) {
    int temp = a % b;
    a = b; b = temp;
  }
  return a;
}
int main() {
  int num1, num2;
  cout << "Enter two numbers: ";
  cin >> num1 >> num2;
  int result = gcd(num1, num2);
  cout << "The GCD of " << num1 << " and " << num2 << " is: " << result
<< endl;

  return 0;
}
```

3. Output: 

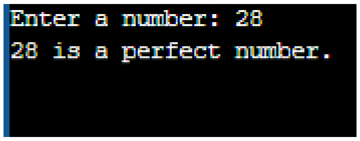
```
Enter two numbers: 43
3
The GCD of 43 and 3 is: 1
```

Problem 7

1. Aim: Write function to check whether the given number is perfect or not.
2. Code:

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

bool isPerfectNumber(int num) {
  if (num <= 0) {
    return false;
  }
  int sum = 1;
  for (int i = 2; i * i <= num; ++i) {
    if (num % i == 0) {
      sum += i;
      if (i * i != num) {
        sum += num / i;
      }
    }
  }
  return sum == num;
}

int main() {
  int number;
  cout << "Enter a number: ";
  cin >> number;
  if (isPerfectNumber(number)) {
    cout << number << " is a perfect number." << endl;
  } else {
    cout << number << " is not a perfect number." << endl;
  }

  return 0;
}
```

3. Output:



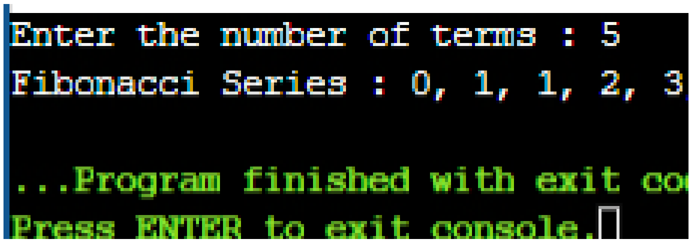
```
Enter a number: 28
28 is a perfect number.
```

Problem 8

1. Aim: To print fabonicci series.
2. Code:

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

int main() {
    int n, first = 0, second = 1, nextTerm = 0;
    cout << "Enter the number of terms : ";
    cin >> n;
    cout << "Fibonacci Series : ";
    for (int i = 1; i <= n; ++i) {
        if(i == 1) {
            cout << first << ", ";
            continue;
        }
        if(i == 2) {
            cout << second << ", ";
            continue;
        }
        nextTerm = first + second;
        first = second;
        second = nextTerm;
        cout << nextTerm << ", ";
    }
    return 0;
}
```

5. Output:



```
Enter the number of terms : 5
Fibonacci Series : 0, 1, 1, 2, 3

...Program finished with exit co
Press ENTER to exit console.
```

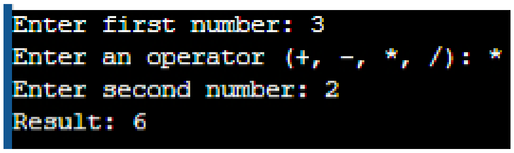
Problem 9

1. Aim: To create a simple calculator.

2. Code:

```
#include <iostream>
using namespace std;
void calculator(double num1, double num2, char operation) {
    switch (operation) {
        case '+':
            cout << "Result: " << num1 + num2 << endl;
            break;
        case '-':
            cout << "Result: " << num1 - num2 << endl;
            break;
        case '*':
            cout << "Result: " << num1 * num2 << endl;
            break;
        case '/':
            if (num2 != 0)
                cout << "Result: " << num1 / num2 << endl;
            else
                cout << "Error: Division by zero is not allowed." << endl;
            break;
        default:
            cout << "Invalid operation. Please use +, -, *, or /." << endl;
    }
}

int main() {
    double num1, num2;
    char operation;
    cout << "Enter first number: ";
    cin >> num1;
    cout << "Enter an operator (+, -, *, /): ";
    cin >> operation;
    cout << "Enter second number: ";
    cin >> num2;
    calculator(num1, num2, operation);
    return 0;
}
```

4. Output:



```
Enter first number: 3
Enter an operator (+, -, *, /): *
Enter second number: 2
Result: 6
```

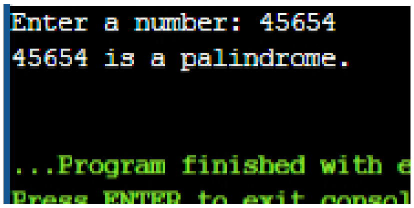
Problem 10

1. Aim: To check the given number is in palindrome or not.

2. Code:

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

bool isPalindrome(int num) {
    int original = num;
    int reversed = 0;
    while (num > 0) {
        int digit = num % 10;
        reversed = reversed * 10 + digit;
        num /= 10;
}
    return original == reversed;
}
int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
    if (isPalindrome(number)) {
        cout << number << " is a palindrome." << endl;
    } else {
        cout << number << " is not a palindrome." << endl;
    }
    return 0;
}
```

4. Output:



```
Enter a number: 45654
45654 is a palindrome.


...Program finished with e
Press ENTER to exit consol
```

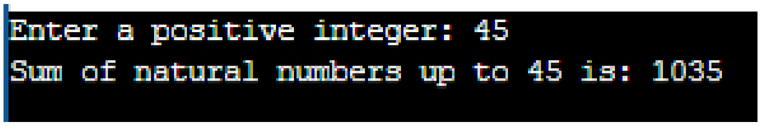
Problem 11

1. Aim: To find the sum of natural numbers using recursion.

2. Code:

```
    #include <iostream>
using namespace std;
int sumOfNaturalNumbers(int n) {
  if (n == 1) {
    return 1;
  }
  else {
    return n + sumOfNaturalNumbers(n - 1);
  }
}
int main() {
  int num;
  cout << "Enter a positive integer: ";
  cin >> num;
  int sum = sumOfNaturalNumbers(num);
  cout << "Sum of natural numbers up to " << num << " is: " << sum << endl;

  return 0;
}
```

3. Output:



```
Enter a positive integer: 45
Sum of natural numbers up to 45 is: 1035
```

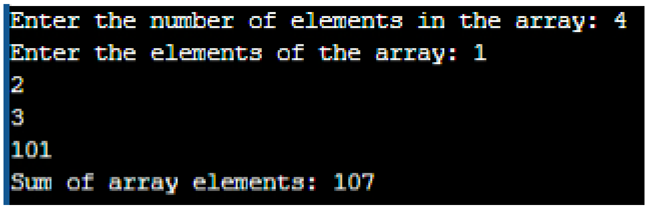
Problem 12

1. Aim: To find the sum of array elements using recursion.

2. Code:

```
        #include <iostream>
using namespace std;
int sumOfArray(int arr[], int n) {
 if (n == 0) return 0;
 return arr[n - 1] + sumOfArray(arr, n - 1);
}
int main() {
 int n;
 cout << "Enter the number of elements in the array: ";
 cin >> n;
 int arr[n];
 cout << "Enter the elements of the array: ";
 for (int i = 0; i < n; ++i) {
 cin >> arr[i];
  }
 int sum = sumOfArray(arr, n);
 cout << "Sum of array elements: " << sum << endl;
 return 0;
}
```

4. Output:



```
Enter the number of elements in the array: 4
Enter the elements of the array: 1
2
3
101
Sum of array elements: 107
```

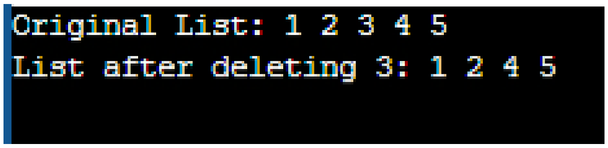
Problem 13

1. Aim: To remove linked list elements.

2. Code:

```
    #include <iostream>
using namespace std;
struct Node {
    int data;
    Node* next;
};
Node* newNode(int data) {
    Node* new_node = new Node;
    new_node->data = data;
    new_node->next = nullptr;
    return new_node;
}
void printList(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}
Node* deleteNode(Node* head, int key) {
    if (head == nullptr) {
        return nullptr;
    }
    if (head->data == key) {
        Node* temp = head;
        head = head->next;
        delete temp;
        return head;
    }
    Node* temp = head;
    while (temp->next != nullptr && temp->next->data != key) {
        temp = temp->next;
    }
    if (temp->next == nullptr) {
        return head;
    }
    Node* to_delete = temp->next;
    temp->next = temp->next->next;
    delete to_delete;

    return head;
}
int main() {
    Node* head = newNode(1);
    head->next = newNode(2);
    head->next->next = newNode(3);
    head->next->next->next = newNode(4);
    head->next->next->next->next = newNode(5);

    cout << "Original List: ";
    printList(head);
    int key = 3;
    head = deleteNode(head, key);
    cout << "List after deleting " << key << ": ";
    printList(head);
    return 0;
}
```

4. Output:



```
Original List: 1 2 3 4 5
List after deleting 3: 1 2 4 5
```

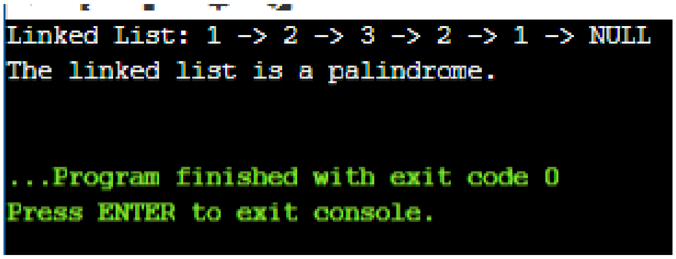
Problem 14

1. Aim: To check if a given linked list is a palindrome.

2. Code:

```
#include <iostream>
#include <stack>
using namespace std;
struct Node {
 int data;
 Node* next;
 Node(int value) : data(value), next(nullptr) {}
};
void append(Node*& head, int value) {
 if (!head) {
 head = new Node(value);
 return;
 }
 Node* temp = head;
 while (temp->next) {
 temp = temp->next;
 }
 temp->next = new Node(value);
}

bool isPalindrome(Node* head) {
  if (!head || !head->next) return true;
  Node* slow = head;
  Node* fast = head;
  stack<int> s;
  while (fast && fast->next) {
    s.push(slow->data);
    slow = slow->next;
    fast = fast->next->next;
  }
  if (fast) {
    slow = slow->next;
  }

  while (slow) {
    if (slow->data != s.top()) return false;
    s.pop();
    slow = slow->next;
  }
  return true;
}
void displayList(Node* head) {
 while (head) {
 cout << head->data << " -> ";
 head = head->next;
 }
 cout << "NULL" << endl;
}
int main() {
 Node* head = nullptr;
 append(head, 1);
 append(head, 2);
 append(head, 3);
 append(head, 2);
 append(head, 1);
 cout << "Linked List: ";
 displayList(head);
 if (isPalindrome(head)) {
 cout << "The linked list is a palindrome." << endl;
 } else {
 cout << "The linked list is not a palindrome." << endl;
 }
 return 0;
}
```

4. Output:



```
Linked List: 1 -> 2 -> 3 -> 2 -> 1 -> NULL
The linked list is a palindrome.


...Program finished with exit code 0
Press ENTER to exit console.
```

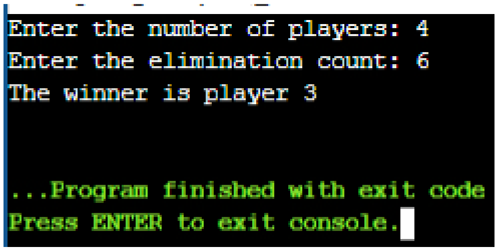
Problem 15

1. Aim: To find the winner of circular game.

2. Code:

```
    #include <iostream>
#include <vector>

using namespace std;

int findTheWinner(int n, int k) {
    vector<int> players(n);
    for (int i = 0; i < n; ++i) {
        players[i] = i + 1;
    }

    int current = 0;
    while (players.size() > 1) {
        current = (current + k - 1) % players.size();
        players.erase(players.begin() + current);
    }

    return players[0];
}

int main() {
    int n, k;
    cout << "Enter the number of players: ";
    cin >> n;
    cout << "Enter the elimination count: ";
    cin >> k;

    int winner = findTheWinner(n, k);
    cout << "The winner is player " << winner << endl;

    return 0;
}
```

4. Output:



```
Enter the number of players: 4
Enter the elimination count: 6
The winner is player 3


...Program finished with exit code
Press ENTER to exit console.
```