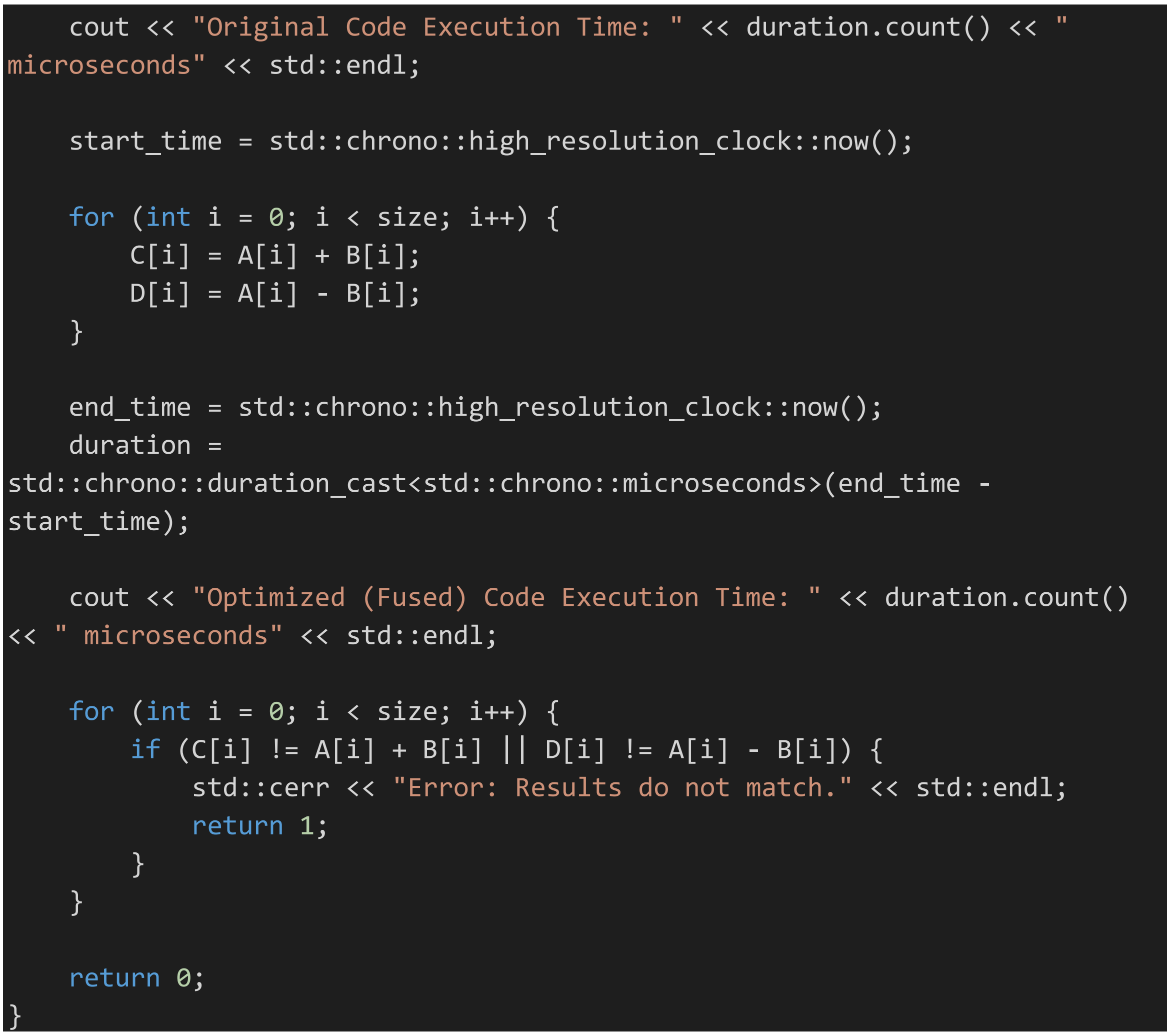
Name: Rachit Shah
Roll No: 20BCE266
Course Name and Course Code: 2CS701 Compiler Construction
Practical No: 9
Aim: To implement Code Optimization techniques: Implement any code optimization technique.

Source Code:

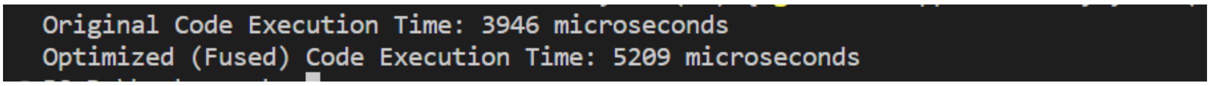
a) **Fusion of loops method:**

```
#include <iostream>
#include <chrono>
#include <bits/stdc++.h>
using namespace std;

int main() {
    const int size = 100000;
    vector<int> A(size);
    vector<int> B(size);
    vector<int> C(size);
    vector<int> D(size);

    for (int i = 0; i < size; i++) {
        A[i] = i;
        B[i] = size - i;
    }

    auto start_time = std::chrono::high_resolution_clock::now();

    for (int i = 0; i < size; i++) {
        C[i] = A[i] + B[i];
    }

    for (int i = 0; i < size; i++) {
        D[i] = A[i] - B[i];
    }

    auto end_time = std::chrono::high_resolution_clock::now();
    auto duration = 
std::chrono::duration_cast<std::chrono::microseconds>(end_time - 
start_time);
```



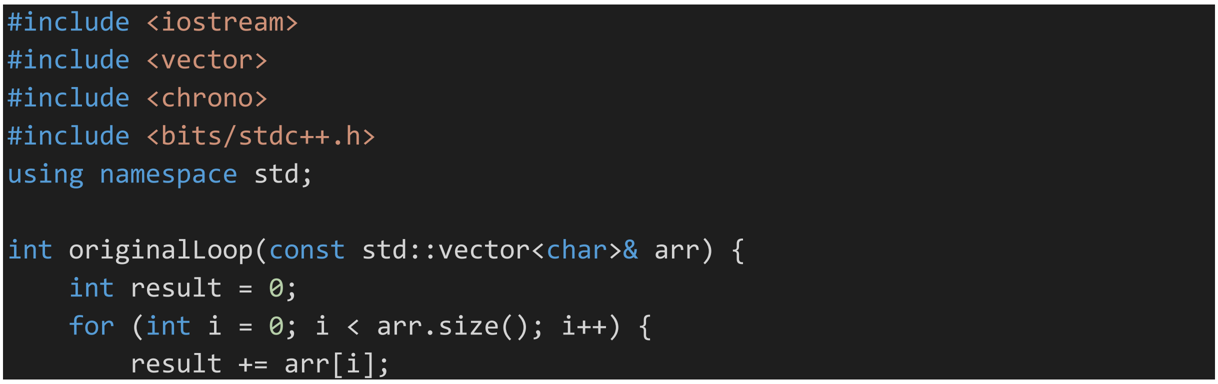
```
    cout << "Original Code Execution Time: " << duration.count() << "
microseconds" << std::endl;

    start_time = std::chrono::high_resolution_clock::now();

    for (int i = 0; i < size; i++) {
        C[i] = A[i] + B[i];
        D[i] = A[i] - B[i];
    }

    end_time = std::chrono::high_resolution_clock::now();
    duration =
std::chrono::duration_cast<std::chrono::microseconds>(end_time -
start_time);

    cout << "Optimized (Fused) Code Execution Time: " << duration.count()
<< " microseconds" << std::endl;

    for (int i = 0; i < size; i++) {
        if (C[i] != A[i] + B[i] || D[i] != A[i] - B[i]) {
            std::cerr << "Error: Results do not match." << std::endl;
            return 1;
        }
    }

    return 0;
}
```

Output:

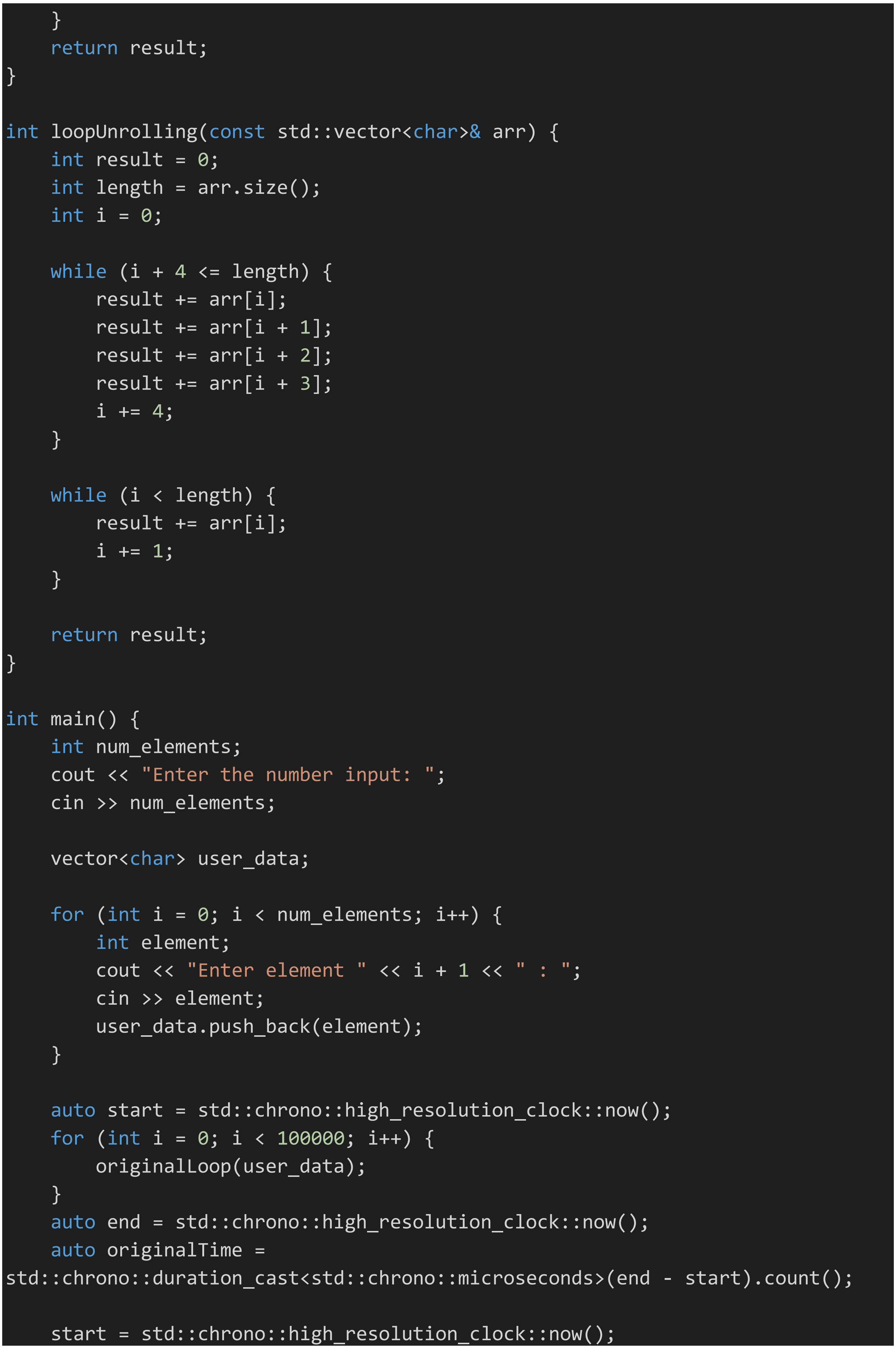


```
Original Code Execution Time: 3946 microseconds
Optimized (Fused) Code Execution Time: 5209 microseconds
```

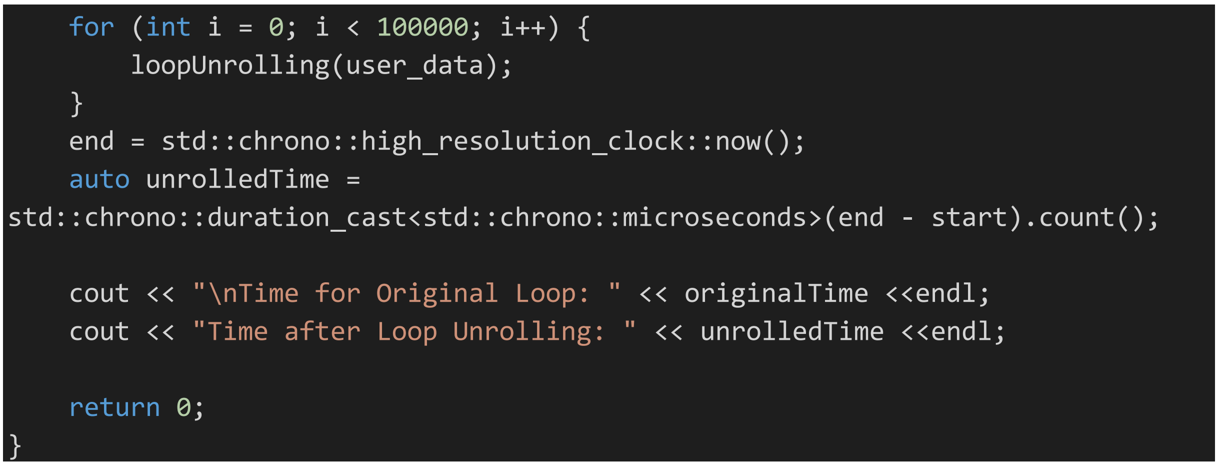
b) Unrolling of loop methos:



```
#include <iostream>
#include <vector>
#include <chrono>
#include <bits/stdc++.h>
using namespace std;

int originalLoop(const std::vector<char>& arr) {
    int result = 0;
    for (int i = 0; i < arr.size(); i++) {
        result += arr[i];
```

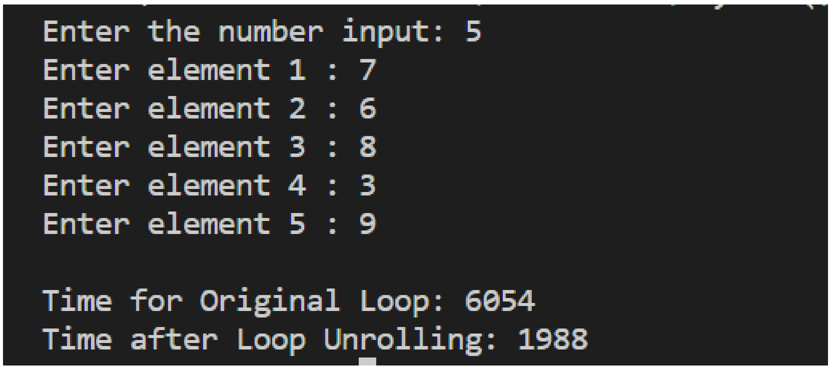


```
    }
    return result;
}

int loopUnrolling(const std::vector<char>& arr) {
    int result = 0;
    int length = arr.size();
    int i = 0;

    while (i + 4 <= length) {
        result += arr[i];
        result += arr[i + 1];
        result += arr[i + 2];
        result += arr[i + 3];
        i += 4;
    }

    while (i < length) {
        result += arr[i];
        i += 1;
    }

    return result;
}

int main() {
    int num_elements;
    cout << "Enter the number input: ";
    cin >> num_elements;

    vector<char> user_data;

    for (int i = 0; i < num_elements; i++) {
        int element;
        cout << "Enter element " << i + 1 << " : ";
        cin >> element;
        user_data.push_back(element);
    }

    auto start = std::chrono::high_resolution_clock::now();
    for (int i = 0; i < 100000; i++) {
        originalLoop(user_data);
    }
    auto end = std::chrono::high_resolution_clock::now();
    auto originalTime =
std::chrono::duration_cast<std::chrono::microseconds>(end - start).count();

    start = std::chrono::high_resolution_clock::now();
```



```
    for (int i = 0; i < 100000; i++) {
        loopUnrolling(user_data);
    }
    end = std::chrono::high_resolution_clock::now();
    auto unrolledTime =
std::chrono::duration_cast<std::chrono::microseconds>(end - start).count();

    cout << "\nTime for Original Loop: " << originalTime <<endl;
    cout << "Time after Loop Unrolling: " << unrolledTime <<endl;

    return 0;
}
```

Output:



```
Enter the number input: 5
Enter element 1 : 7
Enter element 2 : 6
Enter element 3 : 8
Enter element 4 : 3
Enter element 5 : 9

Time for Original Loop: 6054
Time after Loop Unrolling: 1988
```