Assignment: 7

NAME: **Madhuram Brijeshkumar Modi**

ROLL NO.: **21BCP102**

DIV, GROUP: **2, G3**

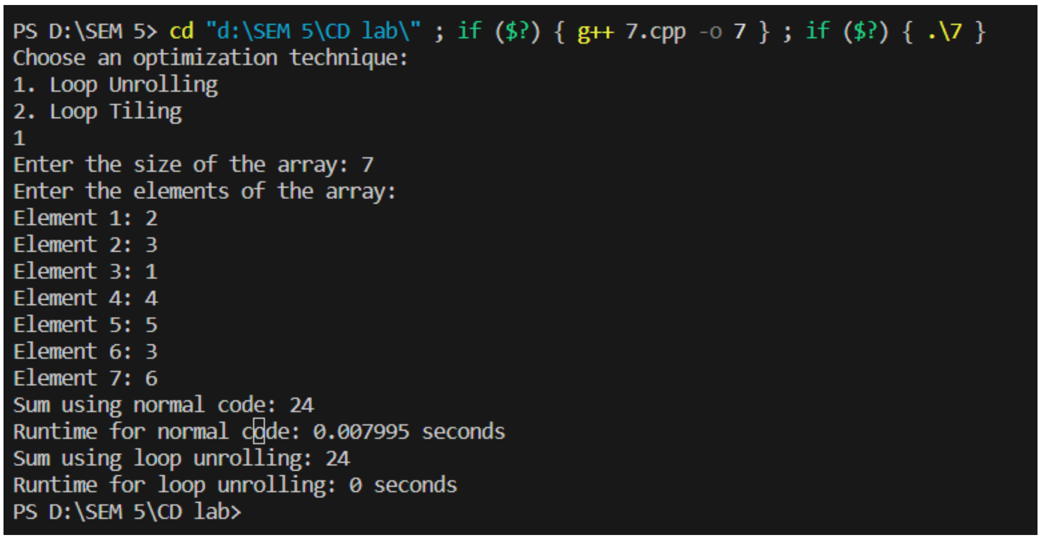
Implement menu driven program to execute any 2 code optimization techniques on given code.

CODE

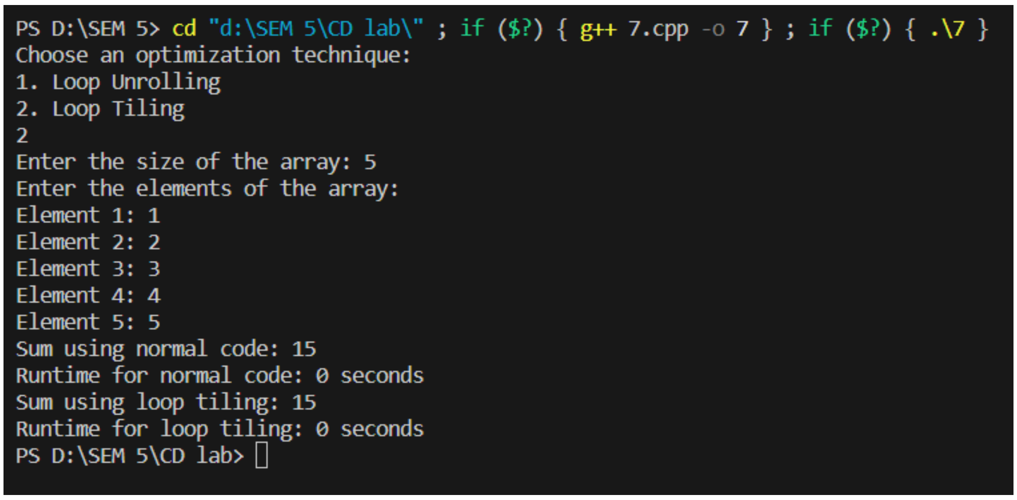
C++:

```cpp
#include <bits/stdc++.h>
#include <chrono> // Function to perform the normal array summation
int sumArrayNormal(int arr[], int size)
{
    int sum = 0;
    for (int i = 0; i < size; i++)
    {
        sum += arr[i];
    }
    return sum;
} // Function to perform array summation with loop unrolling optimization
int sumArrayLoopUnrolling(int arr[], int size)
{
    int sum = 0;
    int i = 0; // Unroll the loop by 4
    for (; i < size - 3; i += 4)
    {
        sum += arr[i] + arr[i + 1] + arr[i + 2] + arr[i + 3];
    } // Handle the remaining elements
    for (; i < size; i++)
    {
        sum += arr[i];
    }
    return sum;
} // Function to perform array summation with loop tiling optimization
int sumArrayLoopTiling(int arr[], int size)
{
    int sum = 0;
    int tile_size = 64; // Set the tile size
    for (int i = 0; i < size; i += tile_size)
    {
        int tile_sum = 0;
        for (int j = i; j < i + tile_size && j < size; j++)
        {
            tile_sum += arr[j];
        }
        sum += tile_sum;
    }
    return sum;
}
int main()
{
    int choice;
    std::cout << "Choose an optimization technique:" << std::endl;
    std::cout << "1. Loop Unrolling" << std::endl;
    std::cout << "2. Loop Tiling" << std::endl;
    std::cin >> choice;
    if (choice != 1 && choice != 2)
    {
        std::cout << "Invalid choice. Please select 1 or 2." << std::endl;
        return 1;
    }
    int size;
    std::cout << "Enter the size of the array: ";
    std::cin >> size;
    if (size <= 0)
    {
        std::cout << "Invalid array size. Please enter a positive integer." << std::endl;
        return 1;
    }
    int arr[size];
    std::cout << "Enter the elements of the array:" << std::endl;
    for (int i = 0; i < size; i++)
    {
        std::cout << "Element " << (i + 1) << ": ";
        std::cin >> arr[i];
    }
    if (choice == 1)
    {
        auto startNormal = std::chrono::high_resolution_clock::now();
        int normalSum = sumArrayNormal(arr, size);
        auto endNormal = std::chrono::high_resolution_clock::now();
        auto startOptimized = std::chrono::high_resolution_clock::now();
        int optimizedSum = sumArrayLoopUnrolling(arr, size);
        auto endOptimized = std::chrono::high_resolution_clock::now();
        std::chrono::duration<double> elapsedNormal = endNormal - startNormal;
        std::chrono::duration<double> elapsedOptimized = endOptimized - startOptimized;
        std::cout << "Sum using normal code: " << normalSum << std::endl;
        std::cout << "Runtime for normal code: " << elapsedNormal.count() << " seconds" <<
std::endl;
        std::cout << "Sum using loop unrolling: " << optimizedSum << std::endl;
        std::cout << "Runtime for loop unrolling: " << elapsedOptimized.count() << "
seconds" << std::endl;
    }
    else if (choice == 2)
    {
        auto startNormal = std::chrono::high_resolution_clock::now();
        int normalSum = sumArrayNormal(arr, size);
        auto endNormal = std::chrono::high_resolution_clock::now();
        auto startOptimized = std::chrono::high_resolution_clock::now();
        int optimizedSum = sumArrayLoopTiling(arr, size);
        auto endOptimized = std::chrono::high_resolution_clock::now();
        std::chrono::duration<double> elapsedNormal = endNormal - startNormal;
        std::chrono::duration<double> elapsedOptimized = endOptimized - startOptimized;
        std::cout << "Sum using normal code: " << normalSum << std::endl;
        std::cout << "Runtime for normal code: " << elapsedNormal.count() << " seconds" <<
std::endl;
        std::cout << "Sum using loop tiling: " << optimizedSum << std::endl;
        std::cout << "Runtime for loop tiling: " << elapsedOptimized.count() << " seconds"
<< std::endl;
    }
    return 0;
}
```

OUTPUT:



```
PS D:\SEM 5> cd "d:\SEM 5\CD lab\" ; if ($?) { g++ 7.cpp -o 7 } ; if ($?) { .\7 }
Choose an optimization technique:
1. Loop Unrolling
2. Loop Tiling
1
Enter the size of the array: 7
Enter the elements of the array:
Element 1: 2
Element 2: 3
Element 3: 1
Element 4: 4
Element 5: 5
Element 6: 3
Element 7: 6
Sum using normal code: 24
Runtime for normal code: 0.007995 seconds
Sum using loop unrolling: 24
Runtime for loop unrolling: 0 seconds
PS D:\SEM 5\CD lab>
```



```
PS D:\SEM 5> cd "d:\SEM 5\CD lab\" ; if ($?) { g++ 7.cpp -o 7 } ; if ($?) { .\7 }
Choose an optimization technique:
1. Loop Unrolling
2. Loop Tiling
2
Enter the size of the array: 5
Enter the elements of the array:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 4
Element 5: 5
Sum using normal code: 15
Runtime for normal code: 0 seconds
Sum using loop tiling: 15
Runtime for loop tiling: 0 seconds
PS D:\SEM 5\CD lab>
```