**Lab Manual: Prefix Sum and Partial Sum in C++**

**Objective**

The goal of this lab is to introduce the concept of prefix and partial sums, enabling students to solve array-based problems involving cumulative sums efficiently.

**Lab Exercises**

**Problem 1: Calculate Prefix Sum Array**

* **Description**: Write a program to generate a prefix sum array from a given array. The prefix sum at each position i is the cumulative sum of elements from the beginning of the array up to position i.
* **Input**: An integer n (size of the array) followed by n integers.
* **Output**: The prefix sum array.

**Code Example**:

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n;

cout << "Enter the number of elements: ";

cin >> n;

vector<int> arr(n), prefix\_sum(n);

cout << "Enter the elements: ";

for (int i = 0; i < n; i++) {

cin >> arr[i];

prefix\_sum[i] = (i == 0) ? arr[i] : prefix\_sum[i - 1] + arr[i];

}

cout << "Prefix Sum Array: ";

for (int i = 0; i < n; i++) {

cout << prefix\_sum[i] << " ";

}

cout << endl;

return 0;

}

**Problem 2: Range Sum Query Using Prefix Sum**

* **Description**: Use a prefix sum array to efficiently calculate the sum of elements within a specific subarray range [l, r].
* **Input**: An integer n for array size, followed by n integers, followed by indices l and r for the range query.
* **Output**: The sum of elements from index l to r (0-based indexing).

**Code Example**:

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n, l, r;

cout << "Enter the number of elements: ";

cin >> n;

vector<int> arr(n), prefix\_sum(n);

cout << "Enter the elements: ";

for (int i = 0; i < n; i++) {

cin >> arr[i];

prefix\_sum[i] = (i == 0) ? arr[i] : prefix\_sum[i - 1] + arr[i];

}

cout << "Enter the range (l and r): ";

cin >> l >> r;

int range\_sum = (l == 0) ? prefix\_sum[r] : prefix\_sum[r] - prefix\_sum[l - 1];

cout << "Sum of elements from index " << l << " to " << r << ": " << range\_sum << endl;

return 0;

}

**Problem 3: Calculate Partial Sum of Even Indices in an Array**

* **Description**: Write a program to compute the partial sum of elements located at even indices in an array.
* **Input**: An integer n (size of the array) followed by n integers.
* **Output**: The sum of elements at even indices.

**Code Example**:

#include <iostream>

#include <vector>

using namespace std;

int main() {

int n, partial\_sum = 0;

cout << "Enter the number of elements: ";

cin >> n;

vector<int> arr(n);

cout << "Enter the elements: ";

for (int i = 0; i < n; i++) {

cin >> arr[i];

if (i % 2 == 0) // Check if the index is even

partial\_sum += arr[i];

}

cout << "Partial Sum of elements at even indices: " << partial\_sum << endl;

return 0;

}

**Problem 4: Prefix Sum for Matrix Row Sums**

* **Description**: Given a 2D matrix of integers, write a program to compute the prefix sum for each row. This will help in efficiently answering queries about row sums in any given sub-range of the row.
* **Input**: Two integers m and n representing matrix dimensions, followed by m rows of n integers each.
* **Output**: Each row's prefix sum array.

**Code Example**:

#include <iostream>

#include <vector>

using namespace std;

int main() {

int m, n;

cout << "Enter the number of rows and columns: ";

cin >> m >> n;

vector<vector<int>> matrix(m, vector<int>(n));

vector<vector<int>> row\_prefix\_sum(m, vector<int>(n));

cout << "Enter the matrix elements:\n";

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cin >> matrix[i][j];

row\_prefix\_sum[i][j] = (j == 0) ? matrix[i][j] : row\_prefix\_sum[i][j - 1] + matrix[i][j];

}

}

cout << "Row-wise Prefix Sum Matrix:\n";

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

cout << row\_prefix\_sum[i][j] << " ";

}

cout << endl;

}

return 0;

}

**Lab Summary**

**In this lab, we covered the calculation of prefix sums and partial sums. Students learned how prefix sums can be used for efficient range sum queries and how partial sums can isolate specific indices for targeted calculations. This lab provides a foundational understanding of cumulative sum techniques in array and matrix** contexts

Task ( part one) :

a)Write a C++ program that uses a for statement to prefix sum a sequence of integers. Assume that the first integer read specifies the number of values remaining to be entered. Your program should read only one value per input statement. A typical input sequence might be

5 100 200 300 400 500

where the 5 indicates that the subsequent 5 values are to be summed.

B) calculate the sum of elements within a specific subarray range [2, 5].

t