

LAB 2 – TIME COMPLEXITY (C2)**Question 1**

- a) Based on the given algorithm, count the number of primitive operations of each step in the algorithm and write the time complexity.

```

Algorithm getSum(A,n)
  Input: An array A storing n integers
  Output: Sum of A
  Sum  $\leftarrow$  0
    for i  $\leftarrow$  0 to n do
      Sum  $\leftarrow$  Sum + A[i]
      increment i
  return Sum

```

Algorithm	Number of primitive operations
Sum \leftarrow 0;	
for i \leftarrow 0 to n do	
Sum \leftarrow Sum + A[i]	
increment i	
return Sum	

- b) Apply the algorithm in a program written in C++/C
- Prompt the user to input the size of the array
 - Then the values for the array
 - Call the function `getSum` (pass the array and the size). Refer to the algorithm.
 - Write the function definition and return the sum.

Question 2

- a) Based on the given algorithm, count the number of primitive operations of each step in the algorithm and write the time complexity.

```

Algorithm arrayAverage (A,n)
  Input: An array A storing n integers
  Output: Average of A

  Line 1: current  $\leftarrow$  0
  Line 2: for i  $\leftarrow$  0 to n-1 do
  Line 3:     current  $\leftarrow$  current + A[i]
  Line 4: return(current/n)

```

Line	Number of primitive operations
1	
2	
3	
4	

b) Apply the algorithm in a program written in C++/C.

Question 3

partialSum(A,S,n) is an algorithm to find the partial sums of an array of n integers.

Algorithm partialSum(A,S,n)
 Input: An array A storing $n \geq 1$ integers
 Output: An array S storing the partial sums of A .

```

S[0] ← A[0]
for i ← 1 to n-1 do
  S[i] ← A[0]
  for j ← 1 to i do
    S[i] ← S[i] + A[j]
```

Here is the sample code in C++ that uses the algorithm. Assume that array $A[8] = \{11, 23, 5, 27, 33, 1, 45, 18\}$.

```

//Lab 2 Q3
#include <iostream>
using namespace std;

int main()
{
    int A [8] = {11,23,5,27,33,1,45,18};
    int S[8];

    S[0] = A[0];
    for(int i=1; i<=7;i++)
    {
        S[i]=A[0];
        for(int j=1; j<=i; j++){
            S[i]= S[i] + A[j];
        }
    }

    cout<<"Array of S is: ";
    for(int i=0 ; i<8; i++)
        cout<<S[i]<<" ";

    return 0;
}
```

- a) What are the values of array **S** after the program is executed? Write the values in the table below.

i	S[i]
0	
1	
2	
3	
4	
5	
6	
7	

- b) What is the time complexity of partialSum? Note: you may write the frequency count.

Algorithm	Number of frequency count
$S[0] \leftarrow A[0]$	
for $i \leftarrow 1$ to $n-1$ do	
$S[i] \leftarrow A[0]$	
for $j \leftarrow 1$ to i do	
$S[i] \leftarrow S[i] + A[j]$	

- c) Based on the code and output in (a), the partialSum algorithm could be improved to **$O(n)$** complexity time. Write the improved partialSum algorithm in the table below, and show the number of primitive operations and time complexity.

Algorithm	Number of primitive operations
$S[0] \leftarrow A[0]$	
for $i \leftarrow 1$ to $n-1$ do	
$S[i] \leftarrow S[i-1] + A[i]$	
increment i	