

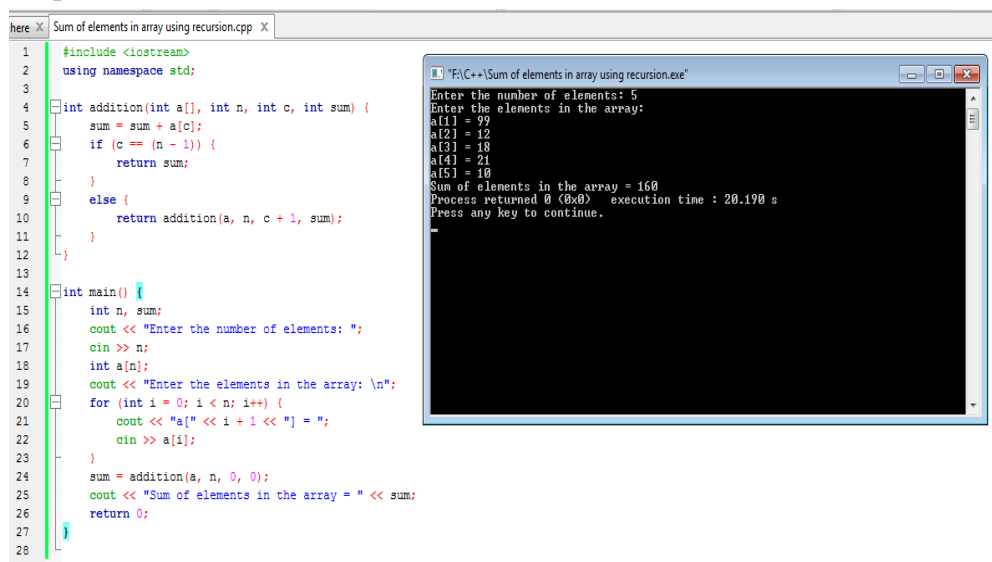
Assignment : 4

Q1) Write a code for addition of elements in the array.

Ans.

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

Output :



The screenshot displays a C++ IDE with two windows. The left window, titled 'Sum of elements in array using recursion.cpp', shows the source code for calculating the sum of array elements using recursion. The right window, titled 'F:\C++\Sum of elements in array using recursion.exe', shows the program's execution. The user enters 5 as the number of elements and provides the array values: 99, 12, 18, 21, and 10. The program outputs the sum of these elements as 160.

```
1 #include <iostream>
2 using namespace std;
3
4 int addition(int a[], int n, int c, int sum) {
5     sum = sum + a[c];
6     if (c == (n - 1)) {
7         return sum;
8     }
9     else {
10        return addition(a, n, c + 1, sum);
11    }
12 }
13
14 int main() {
15     int n, sum;
16     cout << "Enter the number of elements: ";
17     cin >> n;
18     int a[n];
19     cout << "Enter the elements in the array: \n";
20     for (int i = 0; i < n; i++) {
21         cout << "a[" << i + 1 << "] = ";
22         cin >> a[i];
23     }
24     sum = addition(a, n, 0, 0);
25     cout << "Sum of elements in the array = " << sum;
26     return 0;
27 }
28
```

Enter the number of elements: 5
Enter the elements in the array:
a[1] = 99
a[2] = 12
a[3] = 18
a[4] = 21
a[5] = 10
Sum of elements in the array = 160
Process returned 0 (0x0) execution time : 20.190 s
Press any key to continue.

Q2) WAP to perform math operation (+ - * /) using Inline function

Ans

```
#include <iostream>
using namespace std;
inline float calculator(float a, float b, char op) {
    switch (op) {
        case '+':
            return a + b;
        case '-':
            return a - b;
        case '*':
            return a * b;
        case '/':
            if (b != 0) {
                return a / b;
            } else {
                cout << "Division by zero is not possible" << endl;
                return 0;
            }
        default:
            cout << "Invalid operation" << endl;
            return 0;
    }
}

int main() {
    float a, b;
    char op;
    cout << "Enter the first number: ";
    cin >> a;
    cout << "Enter the second number: ";
    cin >> b;
    cout << "Enter the operation (+, -, *, /): ";
    cin >> op;
    float result = calculator(a, b, op);
    cout << "Result: " << result << endl;
    return 0;
}
```

Output :

```
here x WAP to perform math operationusing Inline function.cpp x
1 #include <iostream>
2 using namespace std;
3 inline float calculator(float a, float b, char op) {
4     switch (op) {
5         case '+':
6             return a + b;
7         case '-':
8             return a - b;
9         case '*':
10            return a * b;
11        case '/':
12            if (b != 0) {
13                return a / b;
14            } else {
15                cout << "Division by zero is not possible" << endl;
16                return 0;
17            }
18        default:
19            cout << "Invalid operation" << endl;
20            return 0;
21    }
22 }
23 int main() {
24     float a, b;
25     char op;
26     cout << "Enter the first number: ";
27     cin >> a;
28     cout << "Enter the second number: ";
29     cin >> b;
30     cout << "Enter the operation (+, -, *, /): ";
31     cin >> op;
```

