Programming Practicum Report: Meeting #4

R. Ethan Halim

September 16th, 2024

1 Sum of Series

The entire source file is hosted on a GitHub repository **here**.

1.1 Explanation

The program is to compute the sum of an arithmetic series from 1 to n. Firstly, it requests user input for the value of `n`.

```
int program(std::istream& cin, std::ostream& cout) {
    int64_t n;
    cout << "Input: ";
    cin >> n;

    ...
}
```

$$\text{val} = \sum_{i=1}^{n} i$$

The loop below calculates the sum of the arithmetic series, which is equivalent to the mathematical expression above, and prints the sum.

```
int program(std::istream& cin, std::ostream& cout) {
    ...

    int64_t val = 0;
    for (int64_t i = 1; i <= n; i++) {
        val += i;
```

```
    }

    cout << "Output: " << val << '\n';

    ...
}
```

The explanation to the sum is printed by another loop which at the same time formats the expanded sum of the arithmetic series. The loop iterates from 1 to `n` and prints them following the printing of `"(Explanation:"`. In the iterations from 1 to `n - 1`, `" + "` is appended to the explanation, as to generate something alike `"1 + 2 + ... + [n - 1] + [n] = [val]"`.

```
int program(std::istream& cin, std::ostream& cout) {
    ...

    cout << "(Explanation: ";
    // If it is equal or below zero, just print "0 = 0".
    if (n <= 0) {
        cout << "0";
    }
    for (int64_t i = 1; i <= n; i++) {
        cout << i;
        if (i != n) {
            cout << " + ";
        }
    }
    cout << " = " << val << ")\n";

    return 0;
}
```

1.2 Manual Testing

Below is the compilation and the testing of the source code.

```
avaxar@AvaxarTUF:~/Repos/uni-practica-1/week_4/01_sum$ make
g++ -Wall sum.cpp -o sum
./sum
Input: 15
Output: 120
(Explanation: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 = 120)
```

1.3 Test Cases

1.3.1 Tests

Below is copied directly from the `tests.txt` file.

```
%INPUT
0
%OUTPUT
Input: Output: 0
(Explanation: 0 = 0)
%END

%INPUT
1
%OUTPUT
Input: Output: 1
(Explanation: 1 = 1)
%END

%INPUT
2
%OUTPUT
Input: Output: 3
(Explanation: 1 + 2 = 3)
%END

%INPUT
3
%OUTPUT
Input: Output: 6
(Explanation: 1 + 2 + 3 = 6)
%END

%INPUT
4
%OUTPUT
Input: Output: 10
(Explanation: 1 + 2 + 3 + 4 = 10)
%END

%INPUT
5
%OUTPUT
Input: Output: 15
```

```
(Explanation: 1 + 2 + 3 + 4 + 5 = 15)
%END

%INPUT
10
%OUTPUT
Input: Output: 55
(Explanation: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55)
%END

%INPUT
100
%OUTPUT
Input: Output: 5050
(Explanation: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12
↪  + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 20 + 21 + 22 + 23 + 24
↪  + 25 + 26 + 27 + 28 + 29 + 30 + 31 + 32 + 33 + 34 + 35 + 36
↪  + 37 + 38 + 39 + 40 + 41 + 42 + 43 + 44 + 45 + 46 + 47 + 48
↪  + 49 + 50 + 51 + 52 + 53 + 54 + 55 + 56 + 57 + 58 + 59 + 60
↪  + 61 + 62 + 63 + 64 + 65 + 66 + 67 + 68 + 69 + 70 + 71 + 72
↪  + 73 + 74 + 75 + 76 + 77 + 78 + 79 + 80 + 81 + 82 + 83 + 84
↪  + 85 + 86 + 87 + 88 + 89 + 90 + 91 + 92 + 93 + 94 + 95 + 96
↪  + 97 + 98 + 99 + 100 = 5050)
%END
```

1.3.2 Execution

Below are the results of the test cases. No test cases failed.

```
avaxar@AvaxarTUF:~/Repos/uni-practica-1/week_4/01_sum$ make clean
rm -f sum sum_test
avaxar@AvaxarTUF:~/Repos/uni-practica-1/week_4/01_sum$ make test
g++ -Wall -g sum.cpp -o sum_test -DTEST
./sum_test
[*] The program is currently in test mode!

[*] Running test #1 with the input...
0
[*] Test ran successfully.

[*] Running test #2 with the input...
1
[*] Test ran successfully.

[*] Running test #3 with the input...
2
[*] Test ran successfully.

[*] Running test #4 with the input...
3
[*] Test ran successfully.

[*] Running test #5 with the input...
4
[*] Test ran successfully.

[*] Running test #6 with the input...
5
[*] Test ran successfully.

[*] Running test #7 with the input...
10
[*] Test ran successfully.

[*] Running test #8 with the input...
100
[*] Test ran successfully.

[*] All tests passed.
```

2 Multiplication Table

The entire source file is hosted on a GitHub repository **here**.

2.1 Explanation

The code iterates from 1 to 10 using a for-loop and multiplies the given input `n` by the iterator `i`.

```cpp
int program(std::istream& cin, std::ostream& cout) {
    int64_t n;
    cout << "Input: ";
    cin >> n;

    cout << "\n[Multiplication Table]\n";
    for (int64_t i = 1; i <= 10; i++) {
        cout << n << " x " << i << " = " << (n * i) << '\n';
    }

    return 0;
}
```

2.2 Manual Testing

Below is the compilation and the testing of the source code.

```
avaxar@AvaxarTUF:~/Repos/uni-practica-1/week_4/02_multiplication$ make
g++ -Wall multiplication.cpp -o multiplication
./multiplication
Input: 7

[Multiplication Table]
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

2.3 Test Cases

2.3.1 Tests

Below is copied directly from the `tests.txt` file.

```
%INPUT
0
%OUTPUT
Input:
[Multiplication Table]
0 x 1 = 0
0 x 2 = 0
0 x 3 = 0
0 x 4 = 0
0 x 5 = 0
0 x 6 = 0
0 x 7 = 0
0 x 8 = 0
0 x 9 = 0
0 x 10 = 0
%END

%INPUT
1
%OUTPUT
Input:
[Multiplication Table]
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
%END

%INPUT
2
%OUTPUT
Input:
[Multiplication Table]
```

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
%END

%INPUT
3
%OUTPUT
Input:
[Multiplication Table]
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
%END

%INPUT
4
%OUTPUT
Input:
[Multiplication Table]
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
```

```
%END

%INPUT
5
%OUTPUT
Input:
[Multiplication Table]
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
%END

%INPUT
10
%OUTPUT
Input:
[Multiplication Table]
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
%END

%INPUT
-123
%OUTPUT
Input:
[Multiplication Table]
-123 x 1 = -123
-123 x 2 = -246
-123 x 3 = -369
```

```
-123 x 4 = -492
-123 x 5 = -615
-123 x 6 = -738
-123 x 7 = -861
-123 x 8 = -984
-123 x 9 = -1107
-123 x 10 = -1230
%END
```

2.3.2 Execution

Below are the results of the test cases. No test cases failed.

```
avaxar@AvaxarTUF:~/Repos/uni-practica-1/week_4/02_multiplication$ make clean
rm -f multiplication multiplication_test
avaxar@AvaxarTUF:~/Repos/uni-practica-1/week_4/02_multiplication$ make test
g++ -Wall -g multiplication.cpp -o multiplication_test -DTEST
./multiplication_test
[*] The program is currently in test mode!

[*] Running test #1 with the input...
0
[*] Test ran successfully.

[*] Running test #2 with the input...
1
[*] Test ran successfully.

[*] Running test #3 with the input...
2
[*] Test ran successfully.

[*] Running test #4 with the input...
3
[*] Test ran successfully.

[*] Running test #5 with the input...
4
[*] Test ran successfully.

[*] Running test #6 with the input...
5
[*] Test ran successfully.

[*] Running test #7 with the input...
10
[*] Test ran successfully.

[*] Running test #8 with the input...
-123
[*] Test ran successfully.

[*] All tests passed.
```