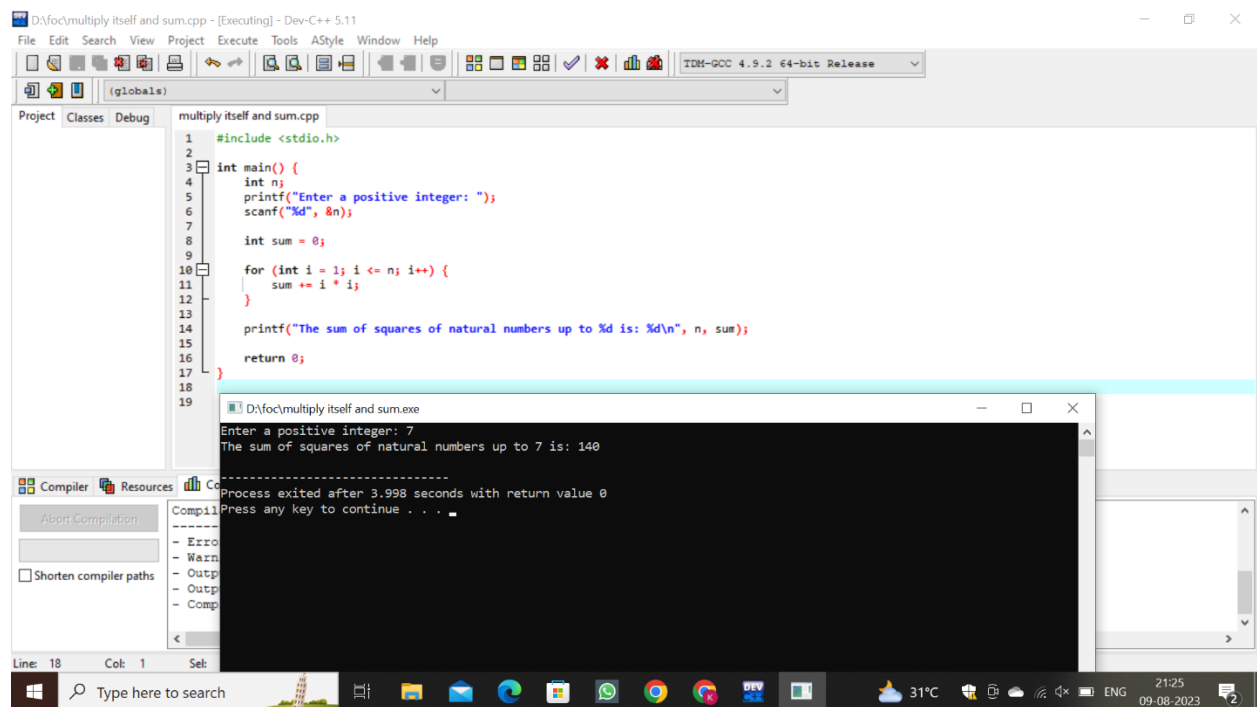


# FOC LAB PRACTICALS DAY 2

T. Sai Krishna

192211870

## 1. summing of no's by squaring of with same no

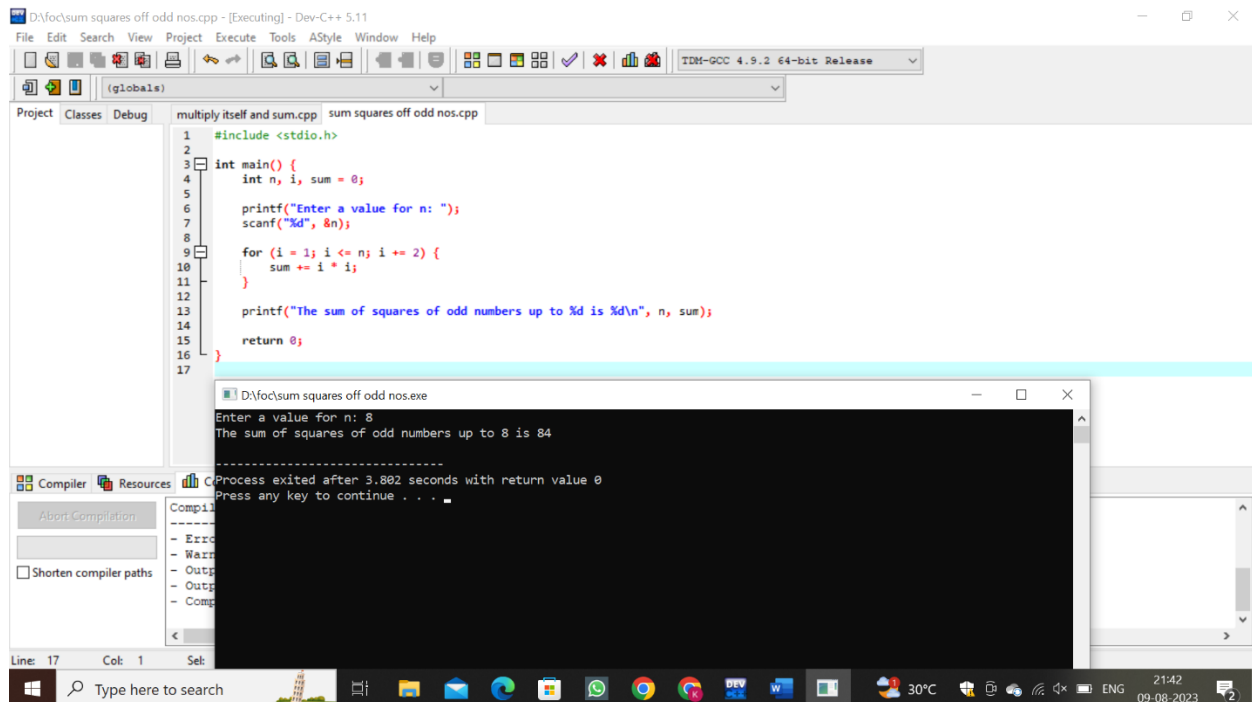


```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     printf("Enter a positive integer: ");
6     scanf("%d", &n);
7
8     int sum = 0;
9
10    for (int i = 1; i <= n; i++) {
11        sum += i * i;
12    }
13
14    printf("The sum of squares of natural numbers up to %d is: %d\n", n, sum);
15
16    return 0;
17 }
18
19
```

Enter a positive integer: 7  
The sum of squares of natural numbers up to 7 is: 140

Process exited after 3.998 seconds with return value 0  
Press any key to continue . . .

## 2.summing of squares of odd no's



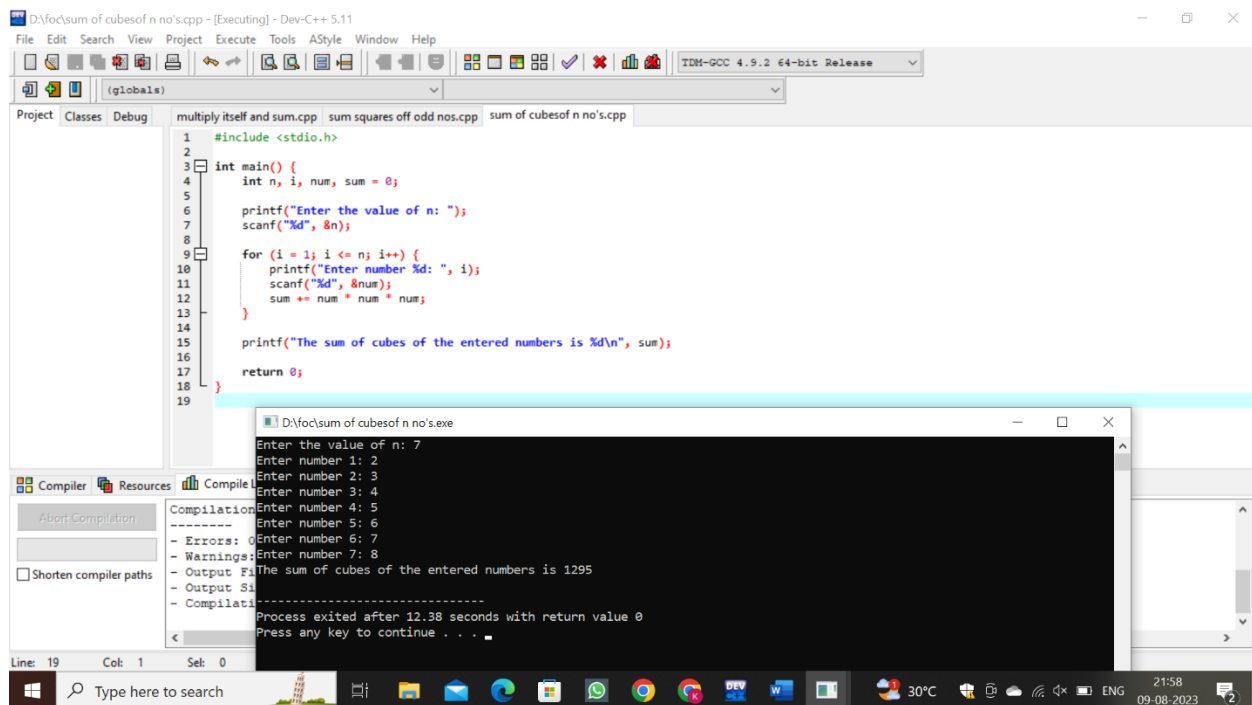
The screenshot shows the Dev-C++ IDE with a project named "sum squares off odd nos.cpp". The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, sum = 0;
5
6     printf("Enter a value for n: ");
7     scanf("%d", &n);
8
9     for (i = 1; i <= n; i += 2) {
10         sum += i * i;
11     }
12
13     printf("The sum of squares of odd numbers up to %d is %d\n", n, sum);
14
15     return 0;
16 }
```

The output window shows the execution results:

```
D:\voc\sum squares off odd nos.exe
Enter a value for n: 8
The sum of squares of odd numbers up to 8 is 84
-----
Process exited after 3.802 seconds with return value 0
Press any key to continue . . .
```

## 3.summing up cubes of n numbers



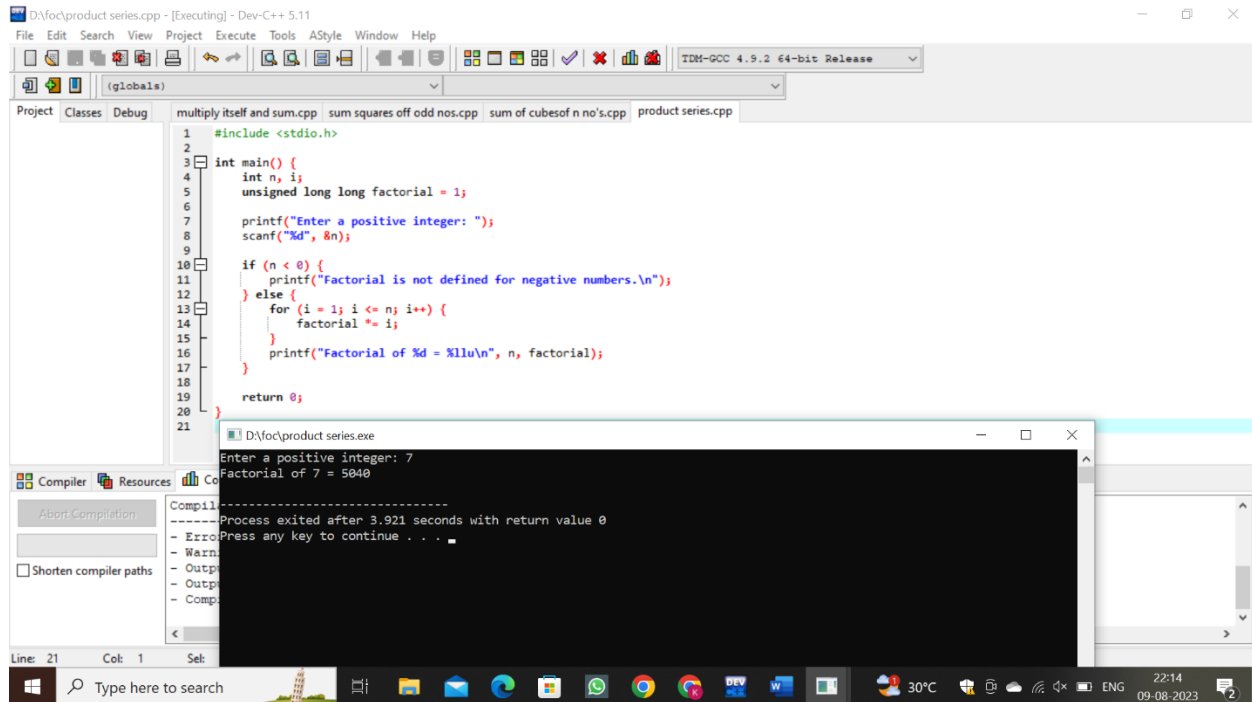
The screenshot shows the Dev-C++ IDE with a project named "sum of cubes of n no's.cpp". The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, num, sum = 0;
5
6     printf("Enter the value of n: ");
7     scanf("%d", &n);
8
9     for (i = 1; i <= n; i++) {
10         printf("Enter number %d: ", i);
11         scanf("%d", &num);
12         sum += num * num * num;
13     }
14
15     printf("The sum of cubes of the entered numbers is %d\n", sum);
16
17     return 0;
18 }
```

The output window shows the execution results:

```
D:\voc\sum of cubes of n no's.exe
Enter the value of n: 7
Enter number 1: 2
Enter number 2: 3
Enter number 3: 4
Enter number 4: 5
Enter number 5: 6
Enter number 6: 7
Enter number 7: 8
The sum of cubes of the entered numbers is 1295
-----
Process exited after 12.38 seconds with return value 0
Press any key to continue . . .
```

## 4.product series



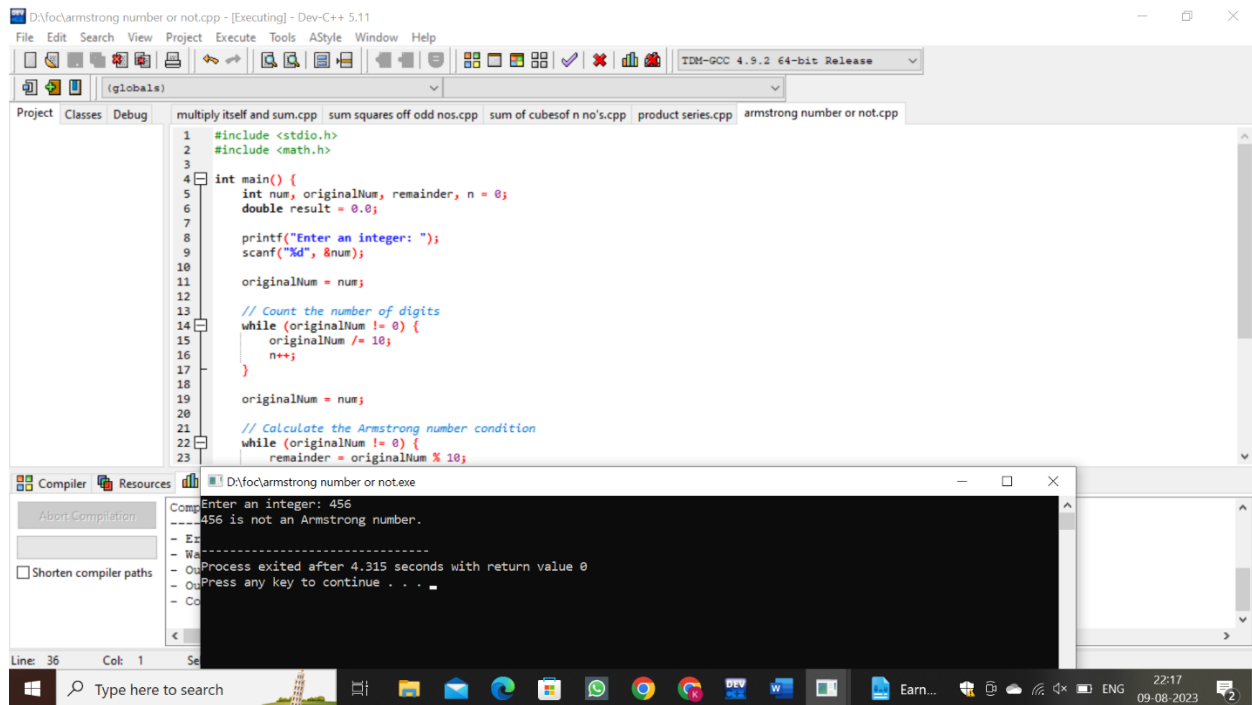
The screenshot shows the Dev-C++ IDE with a C++ program for calculating the factorial of a number. The program is named "product series.cpp" and is located in the "D:\foc\product series.cpp" file. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5     unsigned long long factorial = 1;
6
7     printf("Enter a positive integer: ");
8     scanf("%d", &n);
9
10    if (n < 0) {
11        printf("Factorial is not defined for negative numbers.\n");
12    } else {
13        for (i = 1; i <= n; i++) {
14            factorial *= i;
15        }
16        printf("Factorial of %d = %llu\n", n, factorial);
17    }
18
19    return 0;
20 }
21
```

The program is executed, and the output window shows the following text:

```
D:\foc\product series.exe
Enter a positive integer: 7
Factorial of 7 = 5040
-----
Process exited after 3.921 seconds with return value 0
Press any key to continue . . .
```

## 5.finding given num is Armstrong number or not



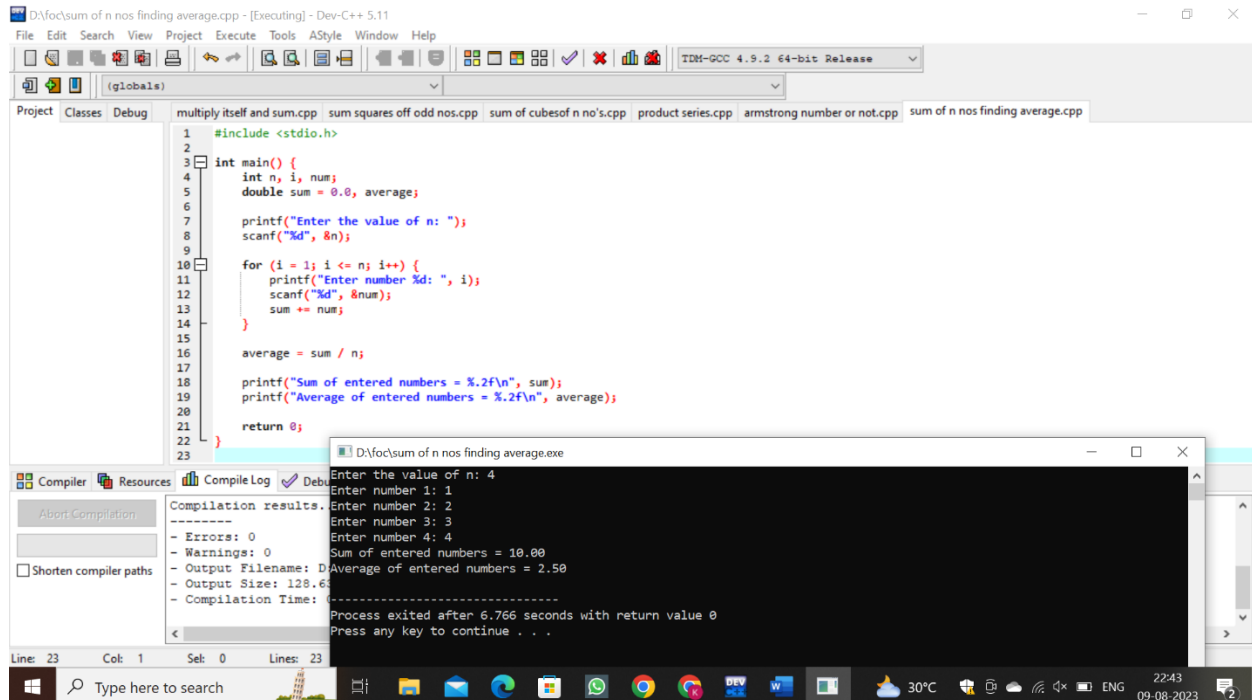
The screenshot shows the Dev-C++ IDE with a C++ program for checking if a number is an Armstrong number. The program is named "armstrong number or not.cpp" and is located in the "D:\foc\armstrong number or not.cpp" file. The code is as follows:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     int num, originalNum, remainder, n = 0;
6     double result = 0.0;
7
8     printf("Enter an integer: ");
9     scanf("%d", &num);
10
11    originalNum = num;
12
13    // Count the number of digits
14    while (originalNum != 0) {
15        originalNum /= 10;
16        n++;
17    }
18
19    originalNum = num;
20
21    // Calculate the Armstrong number condition
22    while (originalNum != 0) {
23        remainder = originalNum % 10;
24        result += pow(remainder, n);
25        originalNum /= 10;
26    }
27
28    if (result == num) {
29        printf("%d is an Armstrong number.\n", num);
30    } else {
31        printf("%d is not an Armstrong number.\n", num);
32    }
33
34    return 0;
35 }
36
```

The program is executed, and the output window shows the following text:

```
D:\foc\armstrong number or not.exe
Enter an integer: 456
456 is not an Armstrong number.
-----
Process exited after 4.315 seconds with return value 0
Press any key to continue . . .
```

## 6. summing up any nos and finding average



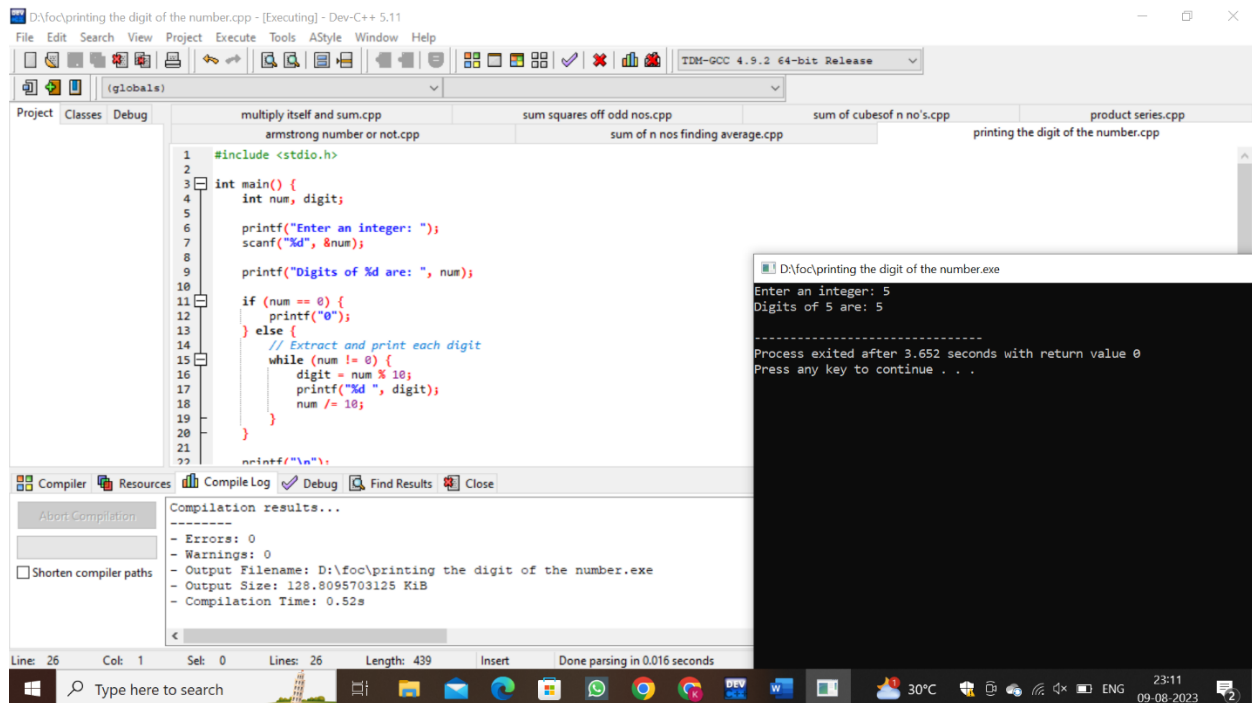
```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, num;
5     double sum = 0.0, average;
6
7     printf("Enter the value of n: ");
8     scanf("%d", &n);
9
10    for (i = 1; i <= n; i++) {
11        printf("Enter number %d: ", i);
12        scanf("%d", &num);
13        sum += num;
14    }
15
16    average = sum / n;
17
18    printf("Sum of entered numbers = %.2f\n", sum);
19    printf("Average of entered numbers = %.2f\n", average);
20
21    return 0;
22 }
23
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\foc\sum of n nos finding average.exe
- Output Size: 128.6 KiB
- Compilation Time: 0.52s

Enter the value of n: 4  
Enter number 1: 1  
Enter number 2: 2  
Enter number 3: 3  
Enter number 4: 4  
Sum of entered numbers = 10.00  
Average of entered numbers = 2.50  
Process exited after 6.766 seconds with return value 0  
Press any key to continue . . .

## 7. printing of the digits of the number



```
1 #include <stdio.h>
2
3 int main() {
4     int num, digit;
5
6     printf("Enter an integer: ");
7     scanf("%d", &num);
8
9     printf("Digits of %d are: ", num);
10
11    if (num == 0) {
12        printf("0");
13    } else {
14        // Extract and print each digit
15        while (num != 0) {
16            digit = num % 10;
17            printf("%d ", digit);
18            num /= 10;
19        }
20
21        printf("\n");
22    }
23}
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: D:\foc\printing the digit of the number.exe
- Output Size: 128.8095703125 KiB
- Compilation Time: 0.52s

Enter an integer: 5  
Digits of 5 are: 5  
Process exited after 3.652 seconds with return value 0  
Press any key to continue . . .

## 8.summing up of digits of an integer

The screenshot shows the Dev-C++ IDE with a C++ project named "sum of digits of integer.cpp". The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int num, digit, sum = 0;
5
6     printf("Enter an integer: ");
7     scanf("%d", &num);
8
9     // Calculate the sum of digits
10    while (num != 0) {
11        digit = num % 10;
12        sum += digit;
13        num /= 10;
14    }
15
16    printf("Sum of digits = %d\n", sum);
17
18    return 0;
19 }
```

The output window shows the execution results:

```
Enter an integer: 4
Sum of digits = 4
-----
Process exited after 5.504 seconds with return value 0
Press any key to continue . . .
```

The compilation results window shows the following details:

- Errors: 0
- Warnings: 0
- Output Filename: D:\foc\sum of di
- Output Size: 128.1298828125 KiB
- Compilation Time: 0.50s

## 9.reversing the digits of an integer

The screenshot shows the Dev-C++ IDE with a C++ project named "reversing digits of integer.cpp". The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4     int num, reversed = 0;
5
6     printf("Enter an integer: ");
7     scanf("%d", &num);
8
9     while (num != 0) {
10        int digit = num % 10;
11        reversed = reversed * 10 + digit;
12        num /= 10;
13    }
14
15    printf("Reversed integer: %d\n", reversed);
16
17    return 0;
18 }
```

The output window shows the execution results:

```
Enter an integer: 5
Reversed integer: 5
-----
Process exited after 2.488 seconds with return value 0
Press any key to continue . . .
```

The compilation results window shows the following details:

- Errors: 0
- Warnings: 0
- Output Filename: D:\foc\reversing digits of integer.cpp
- Output Size: 128.1298828125 KiB
- Compilation Time: 0.50s

## 10.finding the element is odd or even

```
D:\foc\find odd or even.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
multiply itself and sum.cpp sum squares off odd nos.cpp sum of cubes of n no's.cpp product series.cpp armstrong number or not.cpp
sum of n nos finding average.cpp printing the digit of the number.cpp sum of digits of integer.cpp Untitled9 [*] reversing digits of integer.cpp find odd or even.cpp
1 #include <stdio.h>
2
3 int main() {
4     int num;
5
6     printf("Enter an integer: ");
7     scanf("%d", &num);
8
9     if (num % 2 == 0) {
10        printf("%d is an even number.\n", num);
11    } else {
12        printf("%d is an odd number.\n", num);
13    }
14
15    return 0;
16 }
17
Compiler Resources Compile Log
Process exited after 2.934 seconds with return value 0
Press any key to continue . . .
Compilation report
- Errors: 0
- Warnings: 0
- Output Files:
- Output Sizes:
- Compilation:
Shorten compiler paths
Line: 17 Col: 1 Sel: 0
Type here to search
30°C
23:59
09-08-2023
```