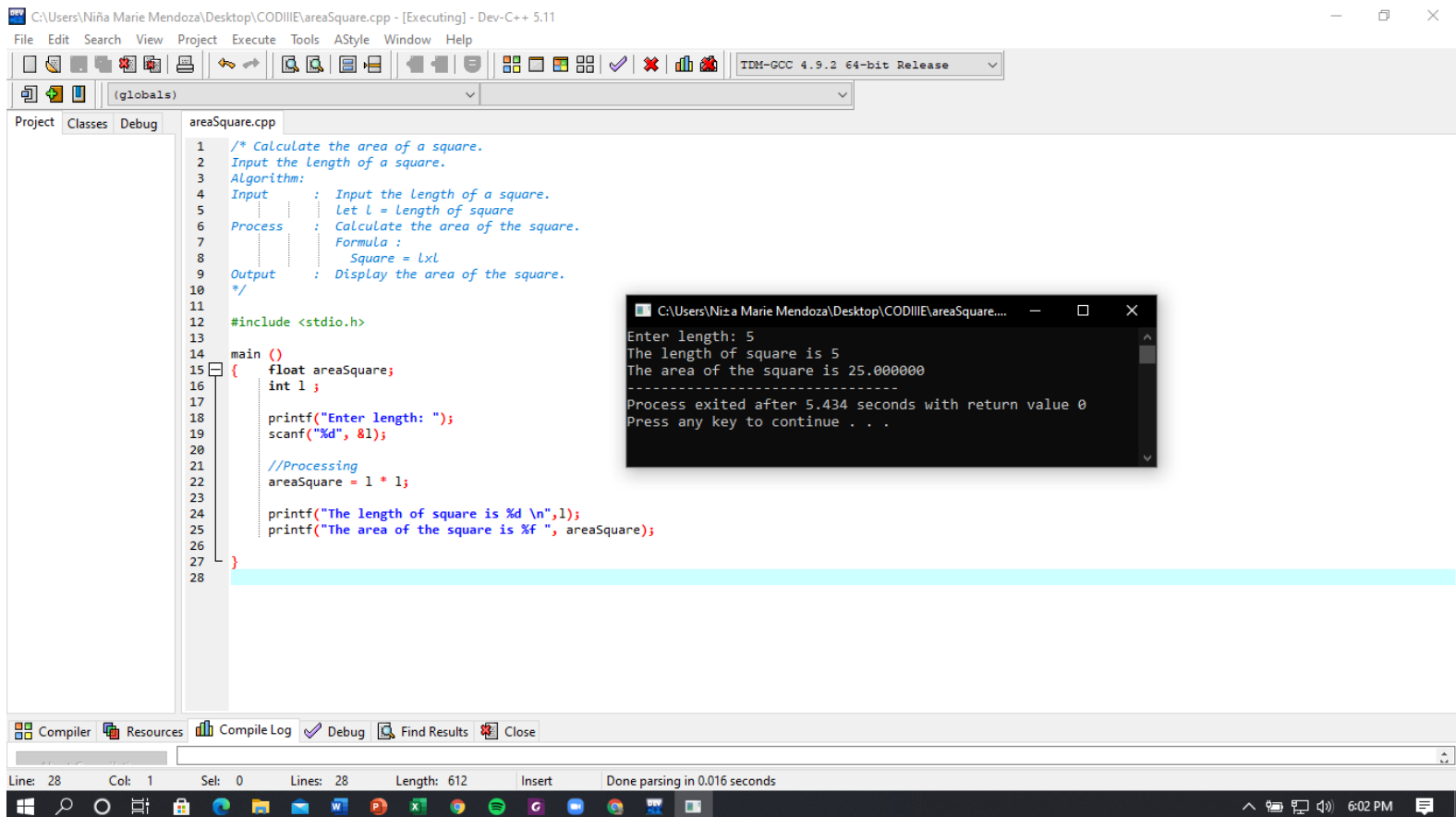


1. Write a program which allows the user to enter the length of a square and which uses this number to calculate and display the area of the square.



```
1  /* Calculate the area of a square.
2  Input the length of a square.
3  Algorithm:
4  Input      : Input the length of a square.
5              : Let l = length of square
6  Process    : Calculate the area of the square.
7              : Formula :
8              : Square = LxL
9  Output     : Display the area of the square.
10 */
11
12 #include <stdio.h>
13
14 main ()
15 {
16     float areaSquare;
17     int l;
18
19     printf("Enter length: ");
20     scanf("%d", &l);
21
22     //Processing
23     areaSquare = l * l;
24
25     printf("The length of square is %d \n", l);
26     printf("The area of the square is %f", areaSquare);
27 }
28
```

Enter length: 5
The length of square is 5
The area of the square is 25.000000

Process exited after 5.434 seconds with return value 0
Press any key to continue . . .

2. Write a program which accepts an amount of money from the user which represents the amount tendered for a purchase. Also, allow the user to enter the cost of the item being purchased. The program should calculate the change due and print this amount. The following could be used as sample output from your program:

Enter the amount tendered: 20.00

Enter the cost of the item: 16.95

The change due is: 3.05

The screenshot displays the Dev-C++ IDE with a C++ program for calculating change. The source code in `changeDue.cpp` includes comments for the algorithm and uses `scanf` for input and `printf` for output. Two terminal windows show sample runs: one with inputs 20 and 16.95 resulting in 3.05 change, and another with inputs 30 and 19.98 resulting in 10.02 change.

```
1  /* Calculate the total change due.
2  Input the amount tendered and the amount purchased.
3  Algorithm:
4  Input: Input the amount tendered and the amount purchased.
5         Let a = amount tendered
6         Let b = amount purchased
7  Process: Calculate the change due.
8  Formula:
9         change = amount tendered - amount purchased
10 Output: Display the total change due.
11 */
12
13 #include <stdio.h>
14
15 main ()
16 {
17     float changeDue, a, b;
18
19     printf("Enter amount tendered: ");
20     scanf("%f", &a);
21
22     printf("Enter amount purchased: ");
23     scanf("%f", &b);
24
25     //processing
26     changeDue = a - b ;
27
28     printf ("The amount tendered is %.2f \n", a);
29     printf ("The amount being purchased is %.2f \n", b);
30     printf ("The total change is %.2f", changeDue);
31 }
32
33
```

Execution Output 1:

```
Enter amount tendered: 20
Enter amount purchased: 16.95
The amount tendered is 20.00
The amount being purchased is 16.95
The total change is 3.05
-----
Process exited after 8.985 seconds with return value 0
Press any key to continue . . .
```

Execution Output 2:

```
Enter amount tendered: 30
Enter amount purchased: 19.98
The amount tendered is 30.00
The amount being purchased is 19.98
The total change is 10.02
-----
Process exited after 7.111 seconds with return value 0
Press any key to continue . . .
```

3. Write a program which can be used to calculate the depth (in metres) of a vertical mine shaft when the time is given for a pebble to fall down the shaft. Use the formula:

$$D = 0.5 * g * t * t$$

Where

- D is the depth of the mine shaft in metres;
- g is the acceleration due to gravity (9.8 m/s); and
- t is the time in seconds.

```
1  /* Calculate the depth (in meters) of the falling pebble into the vertical mine shaft with the given time.
2  Input the value of the given time in seconds.
3  Algorithm:
4  Input: Input the value of the given time in seconds.
5         Let t = time in seconds.
6  Process: Calculate the depth (in meters).
7         formula:
8         D = 0.5 * g * t * t
9  Output: Display the depth of the falling pebble in the vertical mine shaft.
10 */
11
12 #include <stdio.h>
13 main()
14 {
15     float depth, x, g, t;
16
17     x = 0.5;
18     g = 9.8;
19
20     printf("Enter time in seconds: ");
21     scanf("%f", &t);
22
23     //processing
24     depth = x * g * t * t;
25
26     printf("The given time is %.2f s\n", t);
27     printf("The constant is %.2f \n", x);
28     printf("The gravity is %.2f m/s \n", g);
29     printf("The depth of the falling pebble in the vertical mine shaft is %.3f m", depth);
30 }
31
```

Output Window (DEPTH.exe):

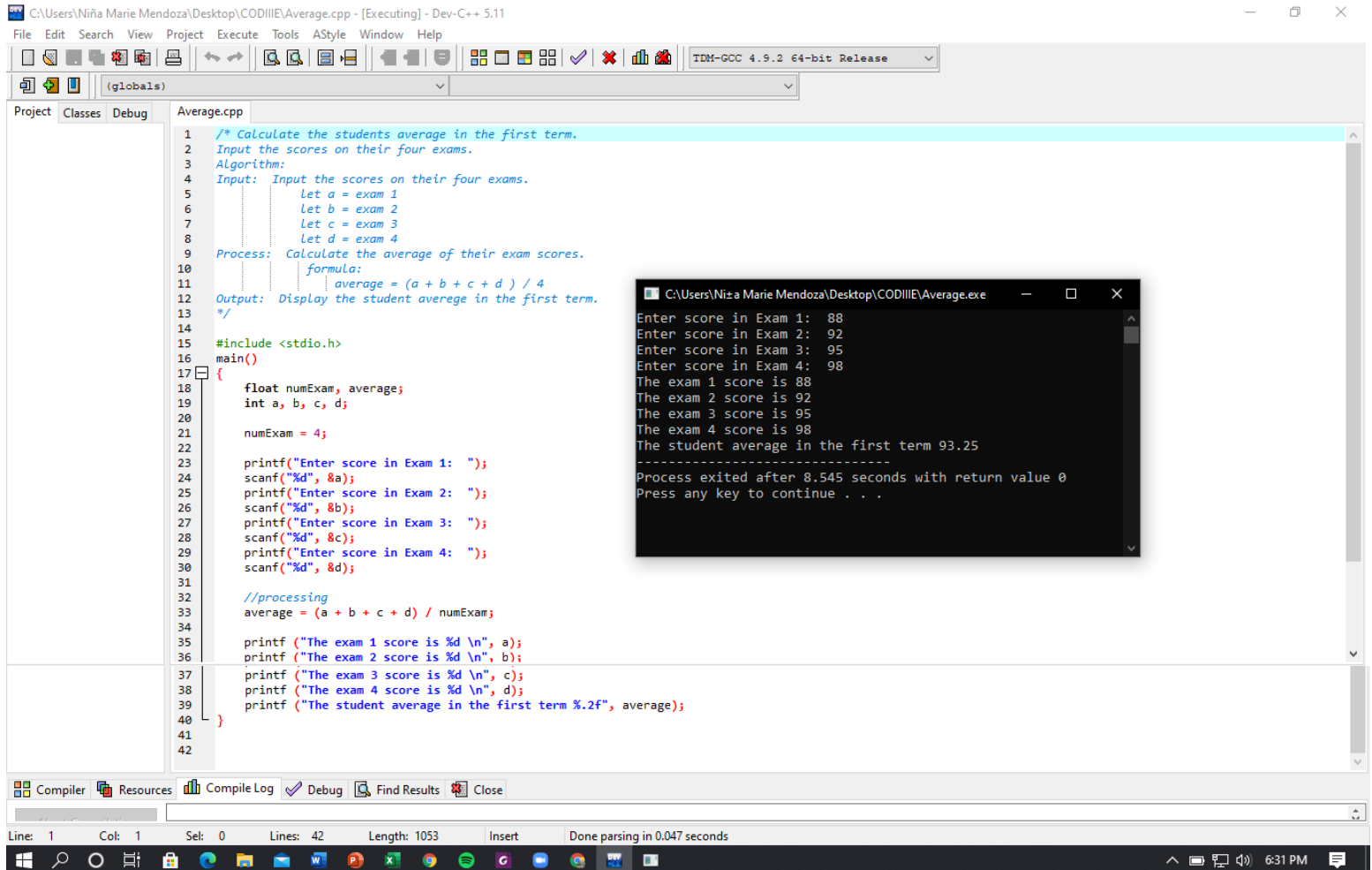
```
Enter time in seconds: 3.5
The given time is 3.50 s
The constant is 0.50
The gravity is 9.80 m/s
The depth of the falling pebble in the vertical mine shaft is 60.025 m
-----
Process exited after 6.944 seconds with return value 0
Press any key to continue . . .
```

4. Write a C program that reads in two integers and then outputs their sum, difference, product and quotient.

```
1  /* Calculate the sum, difference, product, and quotient of a two integers.
2  Input the value of of two integers.
3  Algorithm:
4  Input: Input the value of of two integers.
5  Process: Calculate its sum.
6           Formula: sum = num1 + num2
7           Calculate its difference.
8           Formula: difference = num1 - num2
9           Calculate its product.
10          Formula: product = num1 * num2
11          Calculate its quotient.
12          Formula: quotient = num1 / num2
13 Output: Display the answers.
14 */
15 #include <stdio.h>
16 main ()
17 {
18     int num1, num2, sum, diff, pro, quo;
19
20     printf("Enter num1 = ");
21     scanf("%d", &num1);
22     printf("Enter num2 = ");
23     scanf("%d", &num2);
24
25     //processing
26     sum = num1 + num2;
27     diff = num1 - num2;
28     pro = num1 * num2;
29     quo = num1 / num2;
30
31     printf ("The sum is %d \n", sum);
32     printf ("The difference is %d \n", diff);
33     printf ("The product is %d \n", pro);
34     printf ("The quotient is %d", quo);
35 }
36
```

```
Enter num1 = 25
Enter num2 = 5
The sum is 30
The difference is 20
The product is 125
The quotient is 5
-----
Process exited after 7.928 seconds with return value 0
Press any key to continue . . .
```

5. A class has four exams in one term. Write a program that reads in a student's four exam scores, as integer and outputs the student average.



The screenshot displays the Dev-C++ IDE with the file 'Average.cpp' open. The code is a C++ program designed to calculate the average of four exam scores. It includes comments for each step: calculating the average, inputting scores, and displaying the result. The code uses `scanf` for input and `printf` for output. A console window titled 'C:\Users\Niña Marie Mendoza\Desktop\CODIII\Average.exe' shows the program's execution, where four exam scores (88, 92, 95, 98) are entered, and the resulting average (93.25) is displayed. The program also shows the process exit message.

```
1  /* Calculate the students average in the first term.
2  Input the scores on their four exams.
3  Algorithm:
4  Input: Input the scores on their four exams.
5         Let a = exam 1
6         Let b = exam 2
7         Let c = exam 3
8         Let d = exam 4
9  Process: Calculate the average of their exam scores.
10          formula:
11             average = (a + b + c + d) / 4
12  Output: Display the student average in the first term.
13  */
14
15  #include <stdio.h>
16  main()
17  {
18      float numExam, average;
19      int a, b, c, d;
20
21      numExam = 4;
22
23      printf("Enter score in Exam 1: ");
24      scanf("%d", &a);
25      printf("Enter score in Exam 2: ");
26      scanf("%d", &b);
27      printf("Enter score in Exam 3: ");
28      scanf("%d", &c);
29      printf("Enter score in Exam 4: ");
30      scanf("%d", &d);
31
32      //processing
33      average = (a + b + c + d) / numExam;
34
35      printf("The exam 1 score is %d \n", a);
36      printf("The exam 2 score is %d \n", b);
37      printf("The exam 3 score is %d \n", c);
38      printf("The exam 4 score is %d \n", d);
39      printf("The student average in the first term %.2f", average);
40  }
41
42
```

Execution Output:

```
Enter score in Exam 1: 88
Enter score in Exam 2: 92
Enter score in Exam 3: 95
Enter score in Exam 4: 98
The exam 1 score is 88
The exam 2 score is 92
The exam 3 score is 95
The exam 4 score is 98
The student average in the first term 93.25
-----
Process exited after 8.545 seconds with return value 0
Press any key to continue . . .
```

6. Write a program that reads in two integers into variable X and Y. Extract and display the integer quotient and the remainder of X divided Y.

The screenshot displays the Dev-C++ IDE with a C++ program named `DIVISIONXY.cpp` and its execution output in a separate window.

Source Code (DIVISIONXY.cpp):

```
1  /* Calculate the quotient and the remainder of x divided by y.
2  Input the value of two integers in x and y.
3  Algorithm:
4  Input: Input the value of two integers in x and y.
5  Process: Calculate the quotient.
6          formula:
7          quotient = x / y
8  Calculate the remainder.
9          formula:
10         remainder = x % y
11 Output: Display the quotient and remainder.
12 */
13
14 #include <stdio.h>
15 main()
16 {
17     int x, y, quotient, remainder;
18
19     printf("Enter the value of x: ");
20     scanf("%d", &x);
21     printf("Enter the value of y: ");
22     scanf("%d", &y);
23
24     //processing
25     quotient = x / y;
26     remainder = x % y;
27
28     printf("The value of x is %d \n", x);
29     printf("The value of y is %d \n", y);
30     printf("The quotient of x divided by y is %d \n", quotient);
31     printf("The remainder of x divided by y is %d", remainder);
32
33 }
34
35
```

Execution Output (C:\Users\Niza Marie Mendoza\Desktop\CODIII\DIVISIONXY.exe):

```
Enter the value of x: 20
Enter the value of y: 3
The value of x is 20
The value of y is 3
The quotient of x divided by y is 6
The remainder of x divided by y is 2
-----
Process exited after 3.494 seconds with return value 0
Press any key to continue . . .
```