PRACTICE 4: Flow Control: Repetition (Autonomous Exercises)

THEORIC CONTENTS

- Lesson 2. Algorithms.
- Lesson 3. Data Types.
- Lesson 4. Operators and Expressions.
- Lesson 5. Input and Output
- Lesson 6. Flow Control: Selection
- Lesson 7. Flow Control: Repetition

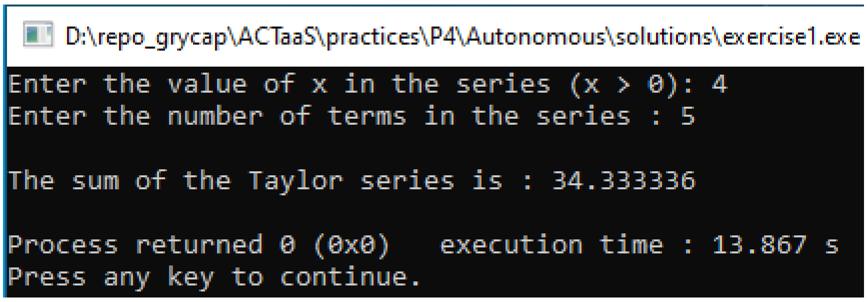
GITHUB CLASSROOM ASSIGNMENT

https://classroom.github.com/a/FWfR7f93

PROPOSED EXERCISES

Exercise 1. Design and implement a program in C that computes the ex-according to the Tailor series:

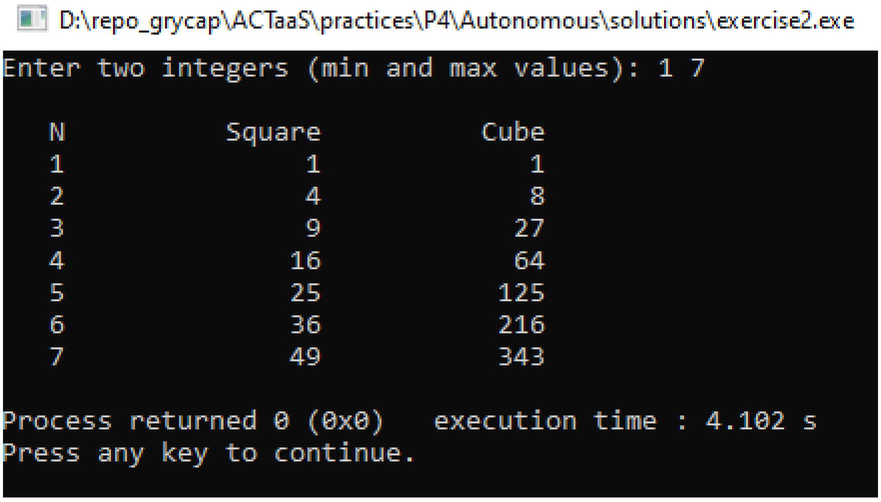
$$e^x = \sum_{i=0}^{N} \frac{x^i}{i!}$$



D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise1.exe

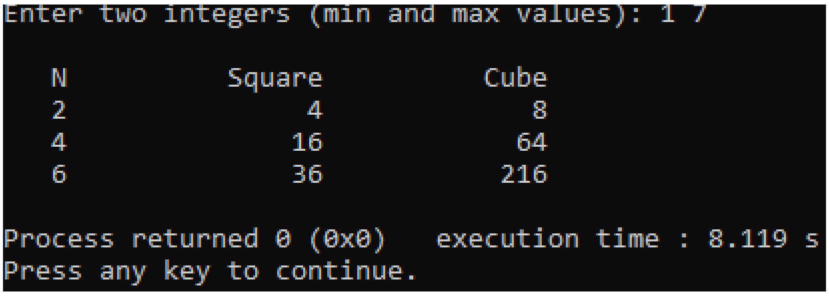
```
Enter the value of x in the series (x > 0): 4
Enter the number of terms in the series : 5

The sum of the Taylor series is : 34.333336

Process returned 0 (0x0)   execution time : 13.867 s
Press any key to continue.
```

Exercise 2. Design and implement a program in C that displays a sequence of integers from an initial value to a final value. The initial and final values must be given by the user. For each integer, in a new line, the program should display the value corresponding to its square and its cube.



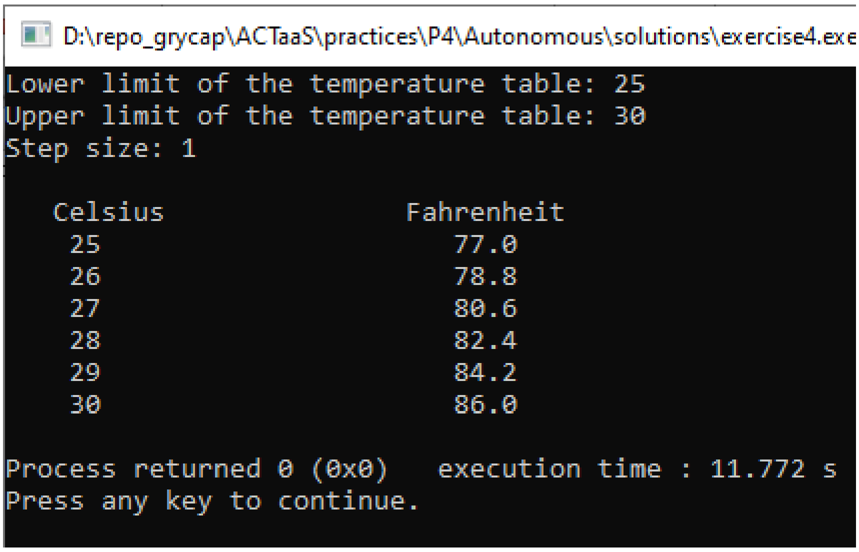
```
D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise2.exe
Enter two integers (min and max values): 1 7

   N          Square          Cube
   1               1             1
   2               4             8
   3               9            27
   4              16            64
   5              25           125
   6              36           216
   7              49           343

Process returned 0 (0x0)   execution time : 4.102 s
Press any key to continue.
```

Exercise 3. Design and implement a program in C that displays a sequence of even integers from an initial value to a final value. The initial and final values must be given by the user. For each integer, in a new line, the program should display the value corresponding to its square and its cube.



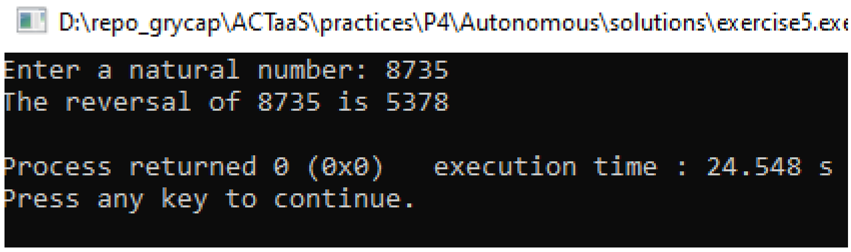
```
Enter two integers (min and max values): 1 7

   N          Square          Cube
   2               4             8
   4              16            64
   6              36           216

Process returned 0 (0x0)   execution time : 8.119 s
Press any key to continue.
```

Exercise 4. Design and implement a program in C to convert Celsius degrees to Fahrenheit. The program should request the starting Celsius value, the ending Celsius value, and the increment. The display should have appropriate headings and list the Celsius value and the corresponding Fahrenheit value. Use the relationship: Fahrenheit = (9.0/5.0)*Celsius + 32.0.



```
D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise4.exe
Lower limit of the temperature table: 25
Upper limit of the temperature table: 30
Step size: 1

   Celsius                Fahrenheit
    25                       77.0
    26                       78.8
    27                       80.6
    28                       82.4
    29                       84.2
    30                       86.0

Process returned 0 (0x0)   execution time : 11.772 s
Press any key to continue.
```

Exercise 5. Design and implement a program in C to reverse the digits of a positive integer number. For example, if the number 8735 is entered, the number displayed should be 5378.



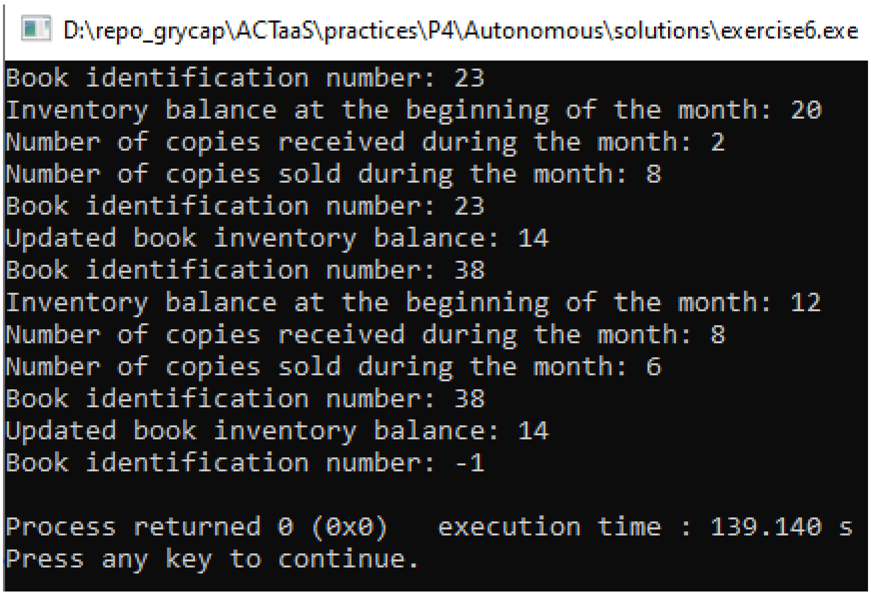
D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise5.ex

```
Enter a natural number: 8735
The reversal of 8735 is 5378

Process returned 0 (0x0)   execution time : 24.548 s
Press any key to continue.
```

Exercise 6 .A bookstore summarizes its monthly transactions by keeping the following information for each book in stock:

Book identification number
Inventory balance at the beginning of the month
Number of copies received during the month
Number of copies sold during the month

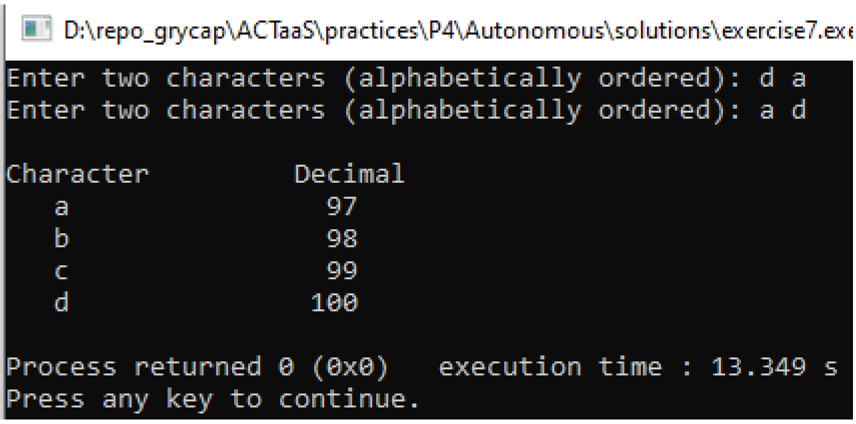
Design and implement a program in C that accepts this data for each book and then displays the book identification number and an updated book inventory balance. The program should end when the user enters a negative book identification number.



D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise6.exe

```
Book identification number: 23
Inventory balance at the beginning of the month: 20
Number of copies received during the month: 2
Number of copies sold during the month: 8
Book identification number: 23
Updated book inventory balance: 14
Book identification number: 38
Inventory balance at the beginning of the month: 12
Number of copies received during the month: 8
Number of copies sold during the month: 6
Book identification number: 38
Updated book inventory balance: 14
Book identification number: -1

Process returned 0 (0x0)   execution time : 139.140 s
Press any key to continue.
```

Exercise 7. Design and implement a program in C that print the decimal values of all characters between the start and stop characters entered by a user. For example, if the user enters an `a` and `z`, the program should print all the characters between `a` and `z` and their respective numerical values. Make sure that the second character entered by the user occurs later in the alphabet than the first character. If it does not a new value for the second character must be read, until the condition is satisfied.

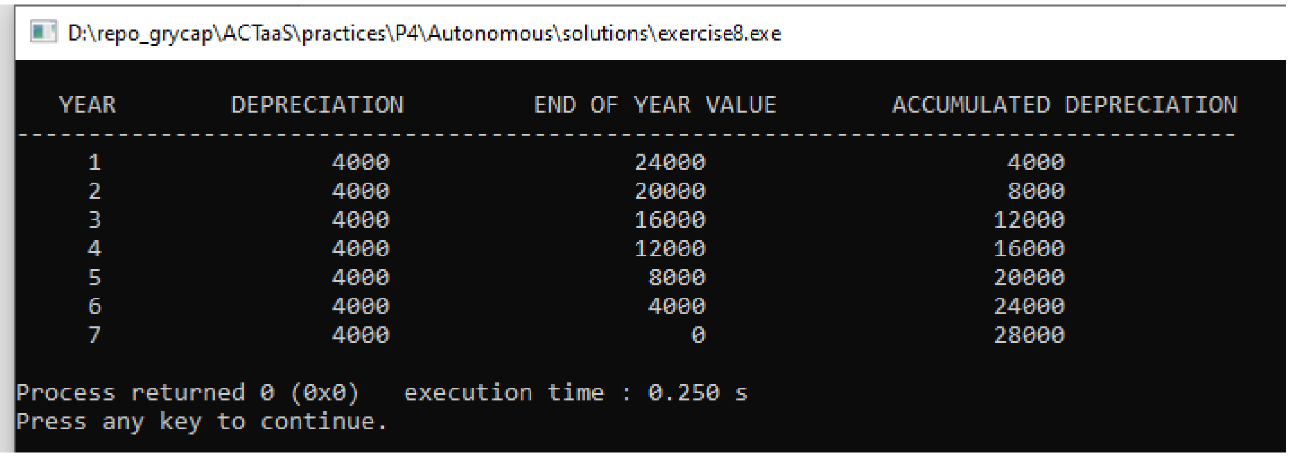

D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise7.exe

```
Enter two characters (alphabetically ordered): d a
Enter two characters (alphabetically ordered): a d

Character          Decimal
   a                 97
   b                 98
   c                 99
   d                100

Process returned 0 (0x0)   execution time : 13.349 s
Press any key to continue.
```

Exercise 8. Design and implement a program in C in which a machine purchased for $28,000 is depreciated at a rate of $4,000 a year for seven years. Write a C program that computes and displays a depreciation table for seven years. The table should have the form:

DEPRECIATION SCHEDULE

		END-OF-YEAR	ACCUMULATED
YEAR	DEPRECIATION	VALUE	DEPRECIATION
1	4000	24000	4000
2	4000	20000	8000
.			
.			
7	4000	0	28000

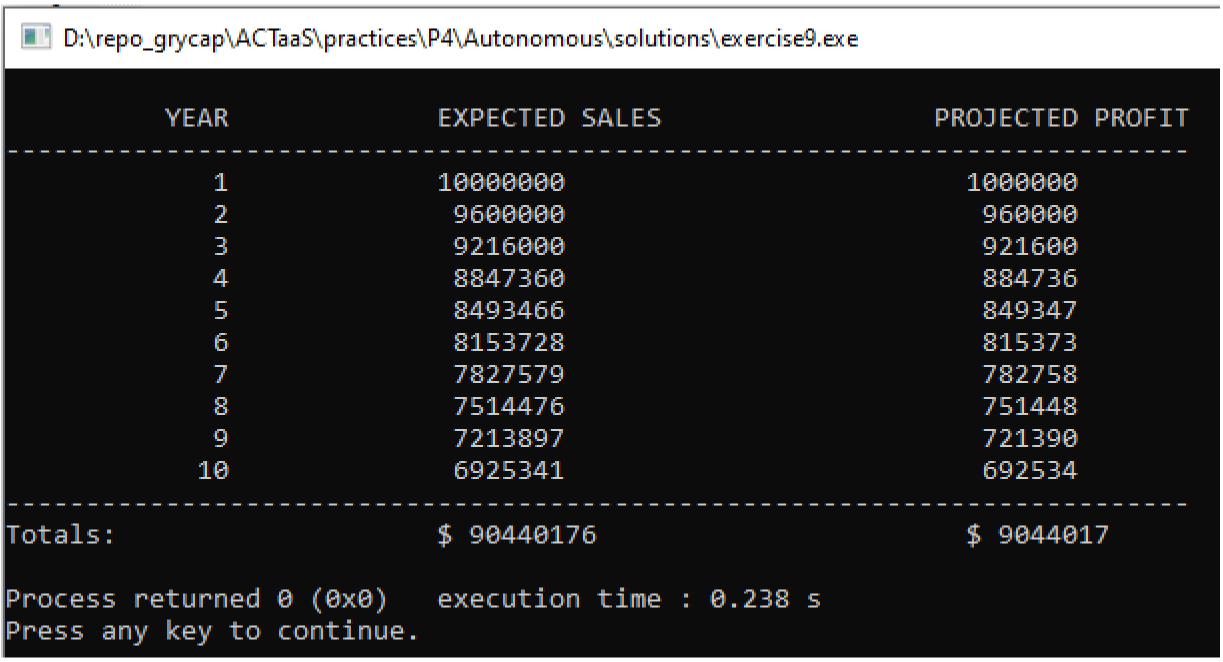

D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise8.exe

```

  YEAR        DEPRECIATION         END OF YEAR VALUE        ACCUMULATED DEPRECIATION
------------------------------------------------------------------------------------
    1                4000                 24000                        4000
    2                4000                 20000                        8000
    3                4000                 16000                       12000
    4                4000                 12000                       16000
    5                4000                  8000                       20000
    6                4000                  4000                       24000
    7                4000                     0                       28000

Process returned 0 (0x0)   execution time : 0.250 s
Press any key to continue.
```

The output of this program must be aligned, according to the previous example. The alignment should be obtained using the printf modifiers.

Exercise 9. A well-regarded manufacturer of widgets has been losing 4 percent of its sales each year. The annual profit for the firm is 10 percent of sales. This year the firm has had $10 million in sales and a profit of $1 million. Determine the expected sales and profit for the next 10 years. Your program should complete and produce display as follows:

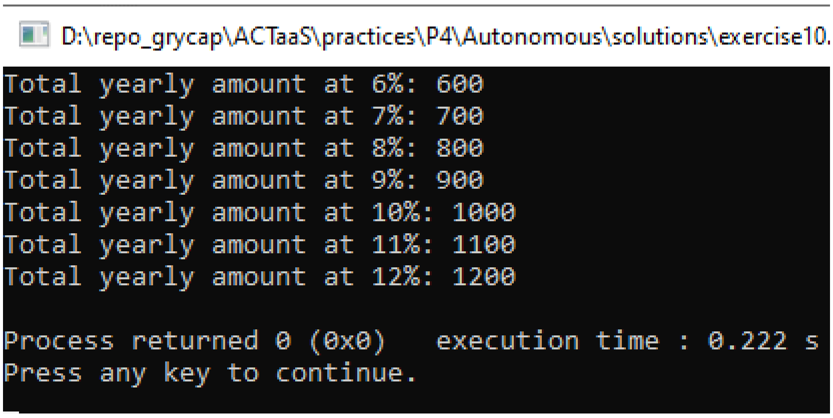
```
SALES AND PROFIT PROJECTION
---------------------------------------------------------------------
YEAR       EXPECTED SALES       PROJECTED PROFIT
---------------------------------------------------------------------
  1        $10000000            $1000000
  2        $  9600000           $  960000
  3
.
.
10
---------------------------------------------------------------------
Totals:    $                    $
```



D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise9.exe

```
          YEAR              EXPECTED SALES                 PROJECTED PROFIT
--------------------------------------------------------------------------
             1              10000000                        1000000
             2               9600000                         960000
             3               9216000                         921600
             4               8847360                         884736
             5               8493466                         849347
             6               8153728                         815373
             7               7827579                         782758
             8               7514476                         751448
             9               7213897                         721390
            10               6925341                         692534
--------------------------------------------------------------------------
Totals:                     $ 90440176                      $ 9044017

Process returned 0 (0x0)   execution time : 0.238 s
Press any key to continue.
```

Exercise 10. Design and implement a program in C that Write a C program that calculates and displays the yearly amount available if $1,000 is invested in a bank account for 10 years, considering different interest rates: from 6% to 12%

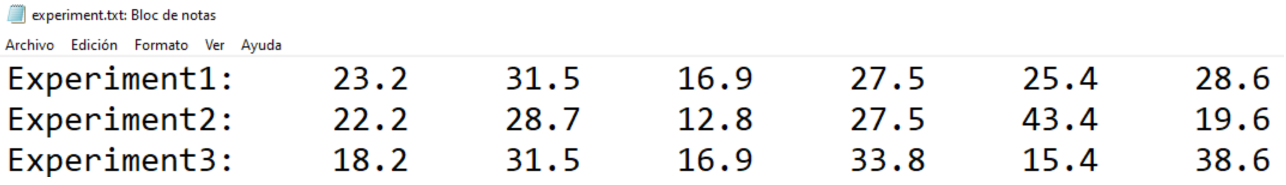

D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise10.

```
Total yearly amount at 6%: 600
Total yearly amount at 7%: 700
Total yearly amount at 8%: 800
Total yearly amount at 9%: 900
Total yearly amount at 10%: 1000
Total yearly amount at 11%: 1100
Total yearly amount at 12%: 1200

Process returned 0 (0x0)   execution time : 0.222 s
Press any key to continue.
```

Exercise 11. Several experiments are performed, each consisting of six test results. The results for each experiment are stored in a file (experiment.txt), which has the following format:

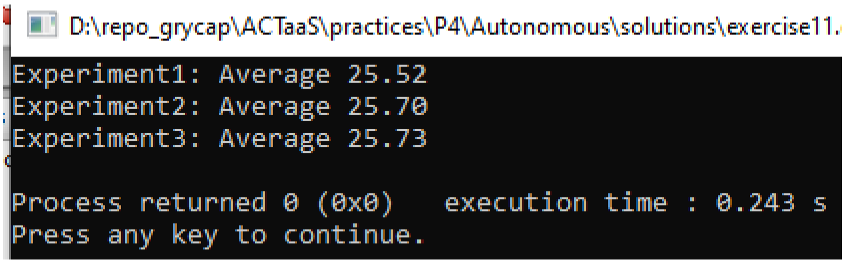
```
Experiment1:   23.2   31.5   16.9   27.5   25.4   28.6
Experiment2:   22.2   28.7   12.8   27.5   43.4   19.6

….
Experimentn:   18.2   31.5   16.9   33.8   15.4   38.6
```

The value of n is unknown.



```
Experiment1:     23.2     31.5     16.9     27.5     25.4     28.6
Experiment2:     22.2     28.7     12.8     27.5     43.4     19.6
Experiment3:     18.2     31.5     16.9     33.8     15.4     38.6
```

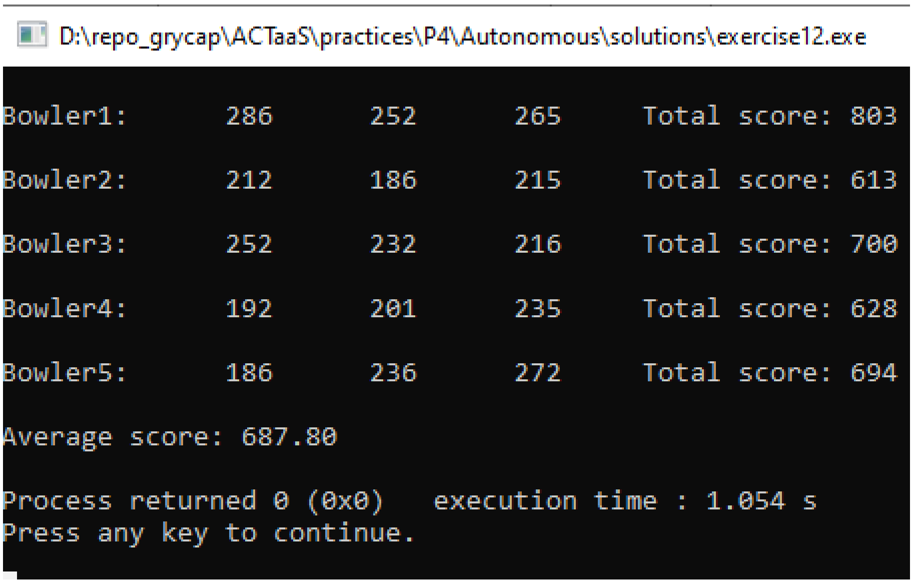
Design and implement a program in C that to compute and display the average of the test results for each experiment.


D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise11.

```
Experiment1: Average 25.52
Experiment2: Average 25.70
Experiment3: Average 25.73

Process returned 0 (0x0)   execution time : 0.243 s
Press any key to continue.
```

Exercise 12. A bowling team consists of five players. Each player bowls three games. Write a C program that displays the total score of each player and the average score, considering the five players. Consider the data are stored in a file (bowling_team.txt), which has the following format.

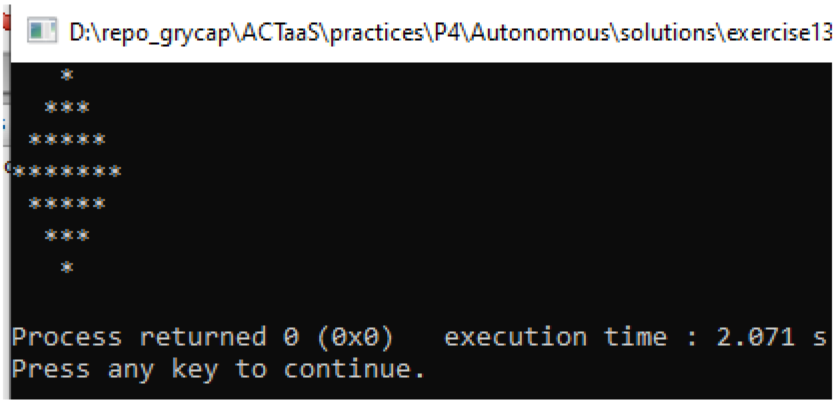
```
Bowler1:   286   252   265
Bowler2:   212   186   215
Bowler3:   252   232   216
Bowler4:   192   201   235
Bowler5:   186   236   272
```



D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise12.exe

```
Bowler1:       286      252      265     Total score: 803

Bowler2:       212      186      215     Total score: 613

Bowler3:       252      232      216     Total score: 700

Bowler4:       192      201      235     Total score: 628

Bowler5:       186      236      272     Total score: 694

Average score: 687.80

Process returned 0 (0x0)   execution time : 1.054 s
Press any key to continue.
```

Exercise 13. Design and implement a program in C a C program to display the following figure:



D:\repo_grycap\ACTaaS\practices\P4\Autonomous\solutions\exercise13

```
   *
  ***
 *****
*******
 *****
  ***
   *


Process returned 0 (0x0)   execution time : 2.071 s
Press any key to continue.
```