***BE 1600***

***Introduction to***

***Programming and Computation***

***Python***

**Assignment 03**

40 points

**Due 10/09/2023 (11:45 A.M.)**

Assignment Objectives:

* To write programs for executing statements repeatedly by using a while loop.
* To develop loops following the loop design strategy.
* To control a loop with a sentinel value.
* To use for loops to implement counter-controlled loops .
* To understand the accumulator pattern

*Solution for this assignment will not be posted on Canvas; however, the solution of any of the assignment problems can be discussed in the class upon request of a student.*

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

1. For non-programming questions, use a word file to type your answers. Don’t use the text box on the Canvas to answer the questions or to write comments, we will not read it. State your answer clearly.
2. For programming questions, include only the source file of each programming problem.
3. Submit your files to the Canvas. You must submit your files on time; otherwise, you will receive zero.
4. Use “Add Another File” feature on Canvas to upload each additional file; do not upload the files as a compressed folder.
5. You can upload your files as many times as you like. Only the last attempt counts because the last files you uploaded are the only files your instructor will see.
6. There will be several modules on the Canvas. You need to upload your files using the correct module on the Canvas.
7. Name each file: *Assignment (assignment number)* for the word file [e.g. Assignment 02] and *Assignment (assignment number) \_ (Question number)* for each programming question [e.g. Assignment 02\_Q03].
8. To upload your file(s):

* In Course Navigation, click the ASSIGNMENTS module.
* Click the title of the assignment.
* Click the **Submit** Assignment button.
* Add **File**. ...
* Add Another **File**. ...
* **Submit** Assignment. ...
* View **Submission**.

*It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.*

***Make sure you review the Cheating & Plagiarism policy on Canvas.***

Write a program for questions Q.1. to Q.9. Convert .py files to text files. Submit 9 files to Canvas by the due date.

**Question 01 (5 points) Done**

Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input **0**. Display the average as a floating-point number rounded to two decimal places. Here are two sample runs:

**Enter an integer, the input ends if it is 0: 4**

**Enter an integer, the input ends if it is 0: 7**

**Enter an integer, the input ends if it is 0: 2**

**Enter an integer, the input ends if it is 0: 3**

**Enter an integer, the input ends if it is 0: 0**

**The number of positives is 4**

**The number of negatives is 0**

**The total is 16**

**The average is 4.00**

**Enter an integer, the input ends if it is 0: 0**

**No numbers are entered except 0**

**Question 02 (4 points)**

Write a program that displays the following table (note that 1 kilogram is 2.2 pounds):

**Kilograms Pounds**

**1 2.2**

**3 6.6**

**...**

**197 433.4**

**199 437.8**

Align all numbers and display pounds as a floating-point number rounded to one decimal place using the format string.

**Question 03 (5 points)**

Write a program that displays the following two tables side by side (note that 1 kilogram is 2.2 pounds and that 1 pound is .45 kilograms):

**Kilograms Pounds | Pounds Kilograms**

**1 2.2 | 20 9.09**

**3 6.6 | 25 11.36**

**...**

**197 433.4 | 510 231.82**

**199 437.8 | 515 235.09**

Align all numbers and display pounds as a floating-point number rounded to one decimal place and kilograms rounded to two decimal places using the format string.

**Question 04 (4 points) Done**

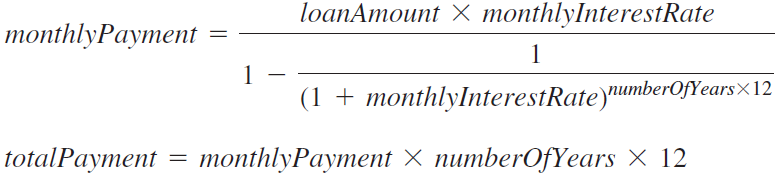
Use a **while** loop to find the largest integer **n** such that **n3** is less than 12,000.

**This number is 22**

**Question 05 (5 points) Decimal problem**

Write a program that lets the user enter the loan amount and loan period in number of years

and displays the monthly and total payments for each interest rate starting from 5% to 8%, with an increment of 1/8. To compute monthly payment, use the below formula.



Here is a sample run:

**Enter loan amount, for example 120000.95: 10000.0**

**Enter number of years as an integer, for example 5: 5**

**Interest Rate Monthly Payment Total Payment**

**5.000% 188.71 11322.74**

**5.125% 189.28 29 11357.13**

**5.250% 189.85 86 11391.59**

**...**

**7.875% 202.17 12129.97**

**8.000% 202.76 12165.83 84**

Format all numbers and use appropriate alignment as shown in the table above using the format string.

**Question 06 (4 points) Done**

Write a program that displays, ten per line, all the leap years in the twenty-first century (from year 2001 to 2100). The years are separated by exactly one space. A leap year is divisible by 4 but not by 100 or divisible by 400.

**2004 2008 2012 2016 2020 2024 2028 2032 2036 2040**

**2044 2048 2052 2056 2060 2064 2068 2072 2076 2080**

**2084 2088 2092 2096**

**Question 07 (4 points) Done**

Write a program that computes the following summation:

****

Use **while loop**. Round the sum to three decimal places.

Here is sample run:

**sum is 24.000**

**Question 08 (4 points) Done**

Write a program to sum the following series:

A number with numbers and a plus and a few numbers

Description automatically generated with medium confidence

Use **for loop**. Round the sum to three decimal places.

Here is sample run:

**sum is 45.124**

**Question 9 (5 points) Done**

Write a program that draws 5 red circles and 5 blue circles with centers (0, 0), as shown in figure below.

Use a loop to print all circles.

