***BE 1600***

***Introduction to***

***Programming and Computation***

***Python***

**Assignment 04**

40 points

**Due 10/23/2023 (11:45 A.M.)**

Assignment Objectives:

* To write nested loops.
* To use random numbers
* To define functions.
* To invoke value-returning functions.
* To invoke functions that does not return a value.

*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. Submit 9 files to Canvas by the due date. **Save all 9 files as text files (.txt).**

**Question 01 (4 points)**

Write a program that displays numbers in a triangular pattern, as shown below. Use **for loops**.

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

**1 2 3 4 5 6**

**1 2 3 4 5 6 7**

**1 2 3 4 5 6 7 8**

**1 2 3 4 5 6 7 8 9**

**1 2 3 4 5 6 7 8 9 10**

**Question 02 (5 points)**

Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period. Use **for loops**. Round all values to two decimal places.

Here a sample run:

Table

Description automatically generated

**Question 03 (4 points)**

Write a **function** named maximum that accepts two integer values as arguments and **returns** the value that is the greater of the two. For example, if 7 and 12 are passed as arguments to the function, the function should return 12. Write a **main function** that prompts the user to enter two integer values and displays the value that is the greater of the two.

Here a sample run:

Enter number 1: 7

Enter number 2: 12

The maximum number is: 12

**Question 04 (5 points)**

Write a **function** that converts milliseconds to hours, minutes, and seconds using the following header: def convertMillis(millis):

The function **returns** a string as hours:minutes:seconds. For example, convertMillis(5500) returns the string 0:0:5, convertMillis(100000) returns the string 0:1:40, and convertMillis(555550000) returns the string 154:19:10.

Write a **main function** that prompts the user to enter a value for milliseconds and displays a string in the format of hours:minutes:seconds.

Here sample runs:

Enter time in milliseconds: 5500

0:0:5

Enter time in milliseconds: 100000

0:1:40

Enter time in milliseconds: 555550000

154:19:10

**Question 05 (5 points)**

Write a **function** that takes parameter i, computes the below series, and **returns** the sum:



Write a **main function** that calls the above function with values 1 to 20 and displays the below table: (Round the values in the second column to four decimal places using format string.)

**i m(i)**

**1 0.5000**

**2 1.1667**

**3 1.9167**

**4 2.7167**

**5 3.5500**

**6 4.4071**

**7 5.2821**

**8 6.1710**

**9 7.0710**

**10 7.9801**

**11 8.8968**

**12 9.8199**

**13 10.7484**

**14 11.6818**

**15 12.6193**

**16 13.5604**

**17 14.5049**

**18 15.4523**

**19 16.4023**

**20 17.3546**

**Question 06 (4 points)**

Write a **function** that **returns** the number of days in a year using the following header:

defnumberOfDaysInAYear(year):

Write a **main function** that displays the number of days in the years from 2000 to 2010

Hint: in a non-leap year, there are 365 days, in a leap year there are 366 days. A year is a leap if it is divisible by 400 or divisible by 4 but not divisible by 100.

Here is a sample run:

**2000 has 366**

**2001 has 365**

**2002 has 365**

**2003 has 365**

**2004 has 366**

**2005 has 365**

**2006 has 365**

**2007 has 365**

**2008 has 366**

**2009 has 365**

**2010 has 365**

**Question 07 (4 points)**

Write a **function** that displays an *n*-by-*n* matrix using the following header:

defprintMatrix(n):

Each element is 0 or 1, which is generated randomly. Write a **main function** that prompts the user to enter **n** and call the above function.

Here is a sample run:

**Enter a number: 5**

**1 0 1 1 0**

**0 1 1 0 0**

**1 0 1 1 1**

**1 1 1 0 1**

**1 0 1 0 0**

**Question 08 (4 points)**

Write a program that generates and prints five random lottery numbers of a three-digit number between 100 and 999. The three digits in the number are the same. Use only one loop and *randint* function to generate the random numbers.

*Note: random library includes functions for working with random numbers; to generate a random integer between a and b, import random library, then use random.randint(a,b) function.*

**Lottery number1: 333**

**Lottery number2: 222**

**Lottery number3: 888**

**Lottery number4: 777**

**Lottery number5: 111**

**Question 9 (5 points)**

Write a program that prompts the user to enter the width and the height of a rectangle. The program displays the rectangle centered at (0, 0) and displays 10 random balls in the rectangle. Use turtle module a while loop to display the 10 random balls.

**Enter the width of rectangle 120**

**Enter the height of rectangle 100**

**Graphical user interface

Description automatically generated with low confidence**