**CSCE 110: Programming I**

**Lab 9**

**General Instructions:**

* The lab is due online by 11:59 pm of the due date. The assignment must be typed, not handwritten, or scanned.
* Make sure you understand everything in this lab before getting started. Also, make sure that your programs match the output exactly as given for each question. This is important as one of the keys to being a good programmer is attention to detail.
* Grading is based on correctness and clarity.
* **Copying work from another source and submitting it as your own is plagiarism and a violation of the code of honor. The minimum penalty for plagiarism is a grade of zero and will be reported to the Aggie honor system office.**

**Lab Questions**

1. In a Python file L9q1.py, write a recursive method, **power**, to compute *x*n by using the following recursive formula:

* *x0* = 1
* *x*n = (*x*n/2)2 if n > 0 and n is even
* *x*n = *x* \* (*x*n/2)2 if n > 0 and n is odd

Write a driver program which will prompt the user to input a real number x and an integer n and uses your **power** method to compute *xn*. If *n* is negative, note that *x*-n = 1/ *xn*, and use that to compute the result correctly. Print the result with an appropriate message.

**Sample Output 1:**

1. Enter a number: 2.5
2. Enter an integer number between -100 and 100: 2
3. 2.5 raised to the power of 2 is 6.25

**Sample Output 2:**

1. Enter a number: -3
2. Enter an integer number between -100 and 100: 3
3. -3.0 raised to the power of 3 is -27.00

**Sample Output 3:**

1. Enter a number: 2
2. Enter an integer number between -100 and 100: 101
3. Invalid input, try again
4. Enter a whole number between -100 and 100: 32
5. 2.0 raised to the power of 32 is 4,294,967,296.00

**Sample Output 4:**

1. Enter a number: 2
2. Enter an integer number between -100 and 100: -2
3. 2.0 raised to the power of -2 is 0.25
4. Ackermann’s Function is a recursive mathematical algorithm that can be used to test how well a system optimizes its performance of recursion. In a Python file L9q2.py, write a recursive method, **ackerman(m,n)** which solves Ackermann’s Function. Use the following logic in your function:

* *If m = 0, then return n + 1*
* *If n = 0, then return ackermann(m - 1, 1)*
* *Otherwise, return ackermann(m - 1, ackermann(m, n - 1))*

**Sample Output 1:**

1. 25 Enter an integer value for m: 0
2. Enter an integer value for n: 3
3. Ackermann(0,3) = 4

**Sample Output 2:**

1. Enter an integer value for m: 2
2. Enter an integer value for n: 0
3. Ackermann(2,0) = 3

**Sample Output 3:**

1. Enter an integer value for m: 2
2. Enter an integer value for n: 3
3. Ackermann(2,3) = 9

**Sample Output 4:**

1. Enter an integer value for m: 3
2. Enter an integer value for n: 4
3. Ackermann(3,4) = 125

**Submitting Your Assignment**

Once you have completed your program, submit your files (L9q1.py, L9q2.py) electronically.

You may resubmit your files as many times as you need until the due date. Only the most recent submission is graded. You are required to include the following lines in the header of all your files:

|  |
| --- |
| **# File: filename.py # Author: Student name # Date: xx/xx/2021 # Section: Student section number  # E-mail: student\_email@tamu.edu  # Description: # e.g. This program asks for ...** |

Submit your files on [gradescope.com](https://www.gradescope.com/)