

Attention:

Submission:

2. Please submit only one zip file including only Python codes with .py extension. There will be a separate .py file for each question in the zip file. Submissions in other formats will not be accepted. Name your Python codes as *lastname1.py*, *lastname2.py*, and so on.
3. Codes are expected to be clear, legible and compact with minimum number of lines and variables, and proper indentation. Variable names are expected to be appropriate, meaningful and self-explanatory. Codes will be graded based on these criteria. A correctly working code does not guarantee full credit.

1. (30 points) Let s be a string that contains a sequence of decimal numbers separated by commas, e.g., $s = '1.23,2.4,3.123'$. Write a Python code that gets s from a user using the **input** statement and prints the sum of the numbers in s . You may use a **while** statement to find the commas in the string, extract one number in each execution of the loop and sum the numbers after all of them are extracted. You may use the built-in string method, **find**.
2. (30 points) Consider the recursive function, $f(x)$, such that $f(0) = \sqrt{\pi}$ and $f(x+1) = (2x+1)f(x)/2$, meaning that $f(1) = \sqrt{\pi}/2$, $f(2) = 3\sqrt{\pi}/4$, etc. Write a Python code that computes this function for a given value of x . The code is expected to get a nonnegative integer, x , from a user using the **input** statement. It is expected to have a recursive function to compute $f(x)$ and print it. Using a regular function instead of a recursive function will cause loss of points. Assume that the user correctly enters a nonnegative integer so that the code does not have to check if it is properly entered.

3. (40 points) Consider the following function:

$$f(x, k) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{k-1} \frac{(-1)^n x^{2n+1}}{n! (2n+1)} = \frac{2}{\sqrt{\pi}} \left[x - \frac{x^3}{3} + \cdots + \frac{(-1)^{k-1} x^{2k-1}}{(k-1)! (2k-1)} \right]$$

Write a Python code that computes this function for a given value of x . The code is expected to get a real number (float), $0 \leq x \leq 1$, and a positive integer, k , from a user using the **input** statement. It is expected to compute $f(x, k)$ and print it. You may directly use the **math.factorial** function to compute the factorials in $f(x, k)$. You may start with $f(x, k) = 0$ and use a for loop with **range(k)** to add each term to $f(x, k) = 0$ in each execution of the loop. Assume that the user enters valid numbers so that the code does not have to check if they are properly entered.