CSC 526 Assignment 4

Q1 (Programming **in PERL**, functions):
The following formula can be used to determine a moving object's kinetic energy:
$KE = ½ mv^2$
KE: kinetic energy;
m: object's mass in kilograms
v: object's velocity, in meters per second
Write a function/subroutine named "kineticEnergy" that accepts the object's mass (in kilograms) and velocity (in meters per second) as arguments. The function/subroutine should return the amount of kinetic energy that the object has. Demonstrate the function/subroutine by calling it in a program that asks the user to enter values for mass and velocity. Provide three different sets of test data.
Estimated time: 30 mins

Q2 (Programming **in PERL,** functions):
A prime number is a number that is only evenly divided by itself and 1.
Write a function/subroutine name "isPrime", which takes an array/list (containing at least 50 different integers) as an argument and returns an array/list of 1s and 0s to indicate if the corresponding integers in the input array/list are prime numbers. Demonstrate the function/subroutine by calling it in a program, the program should print all the integers in the input array/list, and indicate if they are prime numbers or not (by using the output from the function/subroutine).
E.g.
…
5 is a prime number,
6 is not a prime number,
…
Estimated time: 1 hour

Q3 (Programming **in PERL,** more functions and operators):
Choose your favorite song; store the song lyrics in a text file (test input data).
Write a short program that does the following:

- Open the text file, read the content (anyway you choose)
- Display the number of words that the file contains

Estimated time: 30 mins

Q4 (Programming **in PERL,** more functions and operators):
Write a program that asks the user to enter a series of numbers separated by commas. Here is an example of valid input: 17,8,10,12,7,6
The program should calculate and display the sum of all the number. Provide three sets of test data.
Estimated time: 30 mins

Submission instructions:
Please submit a paper copy and an electronic copy.

Paper copy:

- Please submit the paper copy at the beginning of the class.
- Provide (create) **test input data** to all programming questions, and capture the related outputs as screen captures (or output files).
- Print the program source code files, test input data and **the output screen captures (or output files)**. If no output screen capture (or output file) is submitted, it would be assumed that the related program does not compile. If the print-out is not readable, **no mark will be awarded**.
- Identify each assignment question by writing the question number at the top of each page.
- Add the following statement to the first page of your submission: "I have abided by the UNCG Academic Integrity Policy on this assignment". Please write your full name and sign next to the statement. If the statement or the signature is not found, **75% of the possible points will be deducted**.

Electronic copy:

- Please submit a **lastname_firstname_assignment04.zip** (or lastname_firstname_assignment01.rar) file through the Blackboard Digital Dropbox. This zip (or rar) file should contain all submission files.
- Put the answers of all written questions in a **lastname_firstname_assignment04.doc** file.
- For programming questions, you only need to submit the *.pl files.
- Name each *.pl file according to the question number (e.g. Q1_*.pl).

Grading guidelines (programming questions):
Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down into functions / procedures in a reasonable way?
- Style (10%): Is the program indented properly? Do variables have meaningful names?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a comment that identifies the author, the contents, and the compiler used for that particular file? Are all the functions, procedures and data fields clearly documented? Are unclear parts of code documented?

A program that does not compile will get at most **50% of the possible points**.