ENGR 103: Engineering Computation and Algorithmic Thinking

Homework Assignment #4 – Learning Curves

Part 1 – Design

**** DO NOT MODIFY THIS TEMPLATE IN ANY SHAPE OR FORM****
**** SUBMIT THIS DOCUMENT AS A PDF FILE ****

Student's Name: Asher Charlton

Step 1 – Understanding the Problem (7.5 pts)
Use the spaces provided below to answer the following questions:

- (1.5 pts) Do you understand everything in the problem? List anything you do not fully understand.

I understand everything in the problem.

- (2 pts) What are the functional requirements of the program, i.e., what does it need to do?

The program needs to ask the user for input for the first cycle time, then the desired cycle time, and then the slope. While asking for this input it should implement error checking ensuring good input from the user.

- (2 pts) What assumptions are you making?

We are assuming that all input values need to be numeric, that the first cycle time and desired cycle times have to be greater than 0, the first cycle time has to be greater than the goal cycle time, the slope has to be less than 0

- (2 pts) What are the inputs, outputs, etc.?

The input will include the cycle time, the time it takes to complete the first cycle, and the number of the cycles.

The output will include the slope, the cycle number and cycle time, and when a cycle time is less than or equal to the goal cycle time, it will output that the goal has been achieved and will output the learning percent.

Step 2 – Devise a Plan (10 pts)

Use the space provided below to answer the following questions:

- (1.5 pts) What are the decisions that need to be made in this program?

Decisions need to be made about how the calculations will be done, how the user will get input from the user, and where it will store that input.

- (2 pts) What is the sequence of steps you need to complete?

The sequence will start with getting input from the user, then using that input to calculate the slope, then outputting the cycles to up to 100 and then outputting the learning percent along with if the desired cycle time has been achieved, also giving them an option to continue or quit at the end.

- (1.5 pts) How are you going to do the required calculations?

For the required calculations, I'm going to use take their input for the first cycle then multiply that by the number cycle its on to power of the slope. For the learning percent calculations its just 2 to the power of the slope times 100

(5 pts) Based on your answers to the questions in Step 2, provide an algorithm as **pseudocode** or as a **flowchart** showing the specific steps that are needed to create the required computer program. **Be very explicit!!!**

```
PROGRAM Learning Curve Calculator

FUNCTION welcome
   DISPLAY welcome message


FUNCTION goal_cycle_input
   WHILE TRUE LOOP until valid input
      Get goal_cycle from user
      IF not a number OR goal_cycle <= 0 THEN
         DISPLAY error, try again
   END LOOP
   RETURN goal_cycle


FUNCTION first_cycle_input(goal_cycle)
   WHILE TRUE until valid input
      Get first_cycle from user
      IF not a number OR first_cycle <= goal_cycle THEN
         DISPLAY error, try again
   RETURN first_cycle


FUNCTION slope_input
   WHILE TRUE LOOP until valid input
      Get slope from user
      IF not a number OR slope >= 0 THEN
         DISPLAY error, try again
   RETURN slope


FUNCTION calc_cycle_time(first_cycle, cycle, slope)
   RETURN first_cycle * (cycle ^ slope)


FUNCTION number_cycle(first_cycle, slope, goal_cycle)
   cycle = 1

   WHILE TRUE LOOP
      cycle_time = calc_cycle_time(first_cycle, cycle, slope)
      DISPLAY cycle and cycle_time

      IF cycle_time <= goal_cycle THEN
         DISPLAY "goal achieved"
         DISPLAY learning percent (2 ^ slope) * 100

      IF cycle divisible by 100 THEN
         Ask user to continue or quit
         IF quit THEN Exit loop

      cycle = cycle + 1


FUNCTION main
   welcome()
   goal_cycle = goal_cycle_input()
   first_cycle = first_cycle_input(goal_cycle)
   slope = slope_input()
   number_cycle(first_cycle, slope, goal_cycle)
```

Step 4a – Looking Back (7.5 pts)

Create a test plan (on paper) with test cases (i.e., good, bad, and edge cases).

- What are the good, bad, and edge test cases for ALL input in the program?
- What do you expect the results to be in each test case?

Use the table shown below to organize your proposed test cases. Test Case #1 is provided only as an example. You need to provide at least five more test cases. Note: Your test cases must incorporate the values of numerical inputs and numerical results your program will request and/or produce.

If you want to see more test case examples, consult the documents included in the module titled "POLYA 4SDP AND PYTHON STYLE GUIDELINES" in the course's Canvas website.

Test Case #	Value(s)	Expected	Good or Bad?
1	*n = 1,* *test score = 100*	*The average test score should be 100.*	*Good*
2	**Goal = 5** **First = 6** **Slope = -0.500**	**Should go to 2 cycles, 2nd one being 4.874 and the learning percentage should be 81.23%**	**GOOD**
3	**Goal = 0.5** **First = 2** **Slope = -0.100** **Choice = "n"**	**Should go beyond 100 cycles, asking the user at the 100th cycle if they want to continue, and when "n" is selected it will stop the program**	**GOOD**
4	**Goal = "Bad"**	**Will prompt the user to enter in a positive NUMBER again until a correct input is given.**	**BAD**
5	**Goal = 2** **First = 1**	**Will take the goal cycle time and check it against the first cycle time, and since the first cycle time is less than the goal cycle time, it will stop and ask again, I figure its better than just quitting the whole program.**	**BAD**

6	**Goal = 2** **First = 3** **Slope = -0.0001**	**Should take thousands of cycles to get to the goal.**	**EDGE/** **GOOD**
7			
8			
9			