**Practicing Pseudocode**

Write pseudocode for using only these pseudocode phrases for below questions.

* SET <variable>
* PRINT
* COUNT
* GET <variable>
* IF – THEN – ELSE – ENDIF
* FOR – ENDFOR
* WHILE – ENDWHILE
* PROC – ENDPROC – CALL -- RETURN

If you are not sure pseudocode syntax, use the link below for your reference.

Pseudocode Standard: <http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html>.

|  |  |
| --- | --- |
| **Problem** | **Your solution** |
| #1  You are given a word (e.g. “cabbage”) and a letter (e.g. “a”).  - Check if the letter is in the word  - Reply with True or False | SET word  SET letter  IF letter is in word THEN  PRINT True  ELSE   PRINT False ENDIF |
| #2  You are given a word (e.g. “cabbage”) and a letter (e.g. “a”). Use a FOR-loop.   * Count how many times the letter is in the word. * Print with the number of times. | SET word  SET letter SET count to zero  FOR each letter in word  IF current letter is letter THEN  INCREMENT count  ENDIF ENDFOR PRINT count |
| #3  You are not given any information.   * The user must enter a valid letter of the alphabet. * Print the letter if user input a valid letter. | GET value  IF value is a letter THEN  PRINT letter  ENDIF |
| #4  You are given a word (e.g. “cabbage” and a list of letters (e.g. “a”, “b”).   * Check whether all of the letters in the word are in the list. * Print True or False | SET word  SET letter\_list  SET answer to True  FOR letter in word  IF letter NOT in letter\_list  SET answer to False END IF  ENDFOR  PRINT answer |
| 1. Allow the user to input up to 100 positive, one at a time. Keep a running total of the numbers, and print the overall sum. If a negative number is encountered, the problem should terminate, and print the sum so far. Use a WHILE-loop.   SET sum = 0  SET count = 1  GET number  WHILE count <=100 and number >= 0  sum = sum + number  count = count + 1  GET number  END WHILE  PRINT sum | |
| 1. Using nested FOR-loops or WHILE-loops, print an hours and minutes table, of the form:   0 0  0 1  0 2  0 3  ...  0 59  1 0  1 1  1 2  ...  as far as 11 hours 59 mins.  SET hours to zero  WHILE hours < 12  SET minutes to zero  WHILE minutes < 60  Print hours and minutes INCREMENT minutes  END WHILE INCREMENT hours  END WHILE  --------------or--------------  SET hours to zero  FOR each hour from 0 to 11  FOR each minute from 0 to 59  Print hours and minutes  END FOR  END FOR | |
| 1. Input two integer values into variables named start and finish, then print the integers from start to finish inclusive. BUT if start is bigger than finish, swap over their values so they are now in the proper order, then print the integers as specified. Use a procedure named “swap” in your pseudocode for the underlined portion in this description.   PROC swap  temp = start  start = finish  finish = temp  ENDPROC  GET start and finish  IF start > finish  CALL swap  ENDIF  FOR num in range start to finish  PRINT num  ENDFOR | |