COS 125 F16 HW #2 Due Friday Sept 23, 2016 EoD

Rev 9/15/2016 - Examples in #4 corrected

SUBMIT YOUR HOMEWORK IN A SINGLE PDF FILE ON BLACKBOARD.

For each programming problem include a listing of the code along with the resulting output. Since you are submitting electronically, please include the complete output. Because all of these programs are single-file Python programs, you can simply paste your source code into a document and likewise for the output. `Use a monospaced font such as Consolas, Courier New, or Source Code Pro for program code and problem 5 output.` If using Word, you might also alter the default paragraph spacing to 0 both before and after the "paragraph."

This homework assignment uses loops and iteration, conditional statements, and some Python functions such as chr() and ord() as well as type() and/or type conversion functions. **This is a very important assignment in terms of learning to program using iteration and selection.** As mentioned in class, problems 3 and 4 could appear on an exam if they had not already been assigned here.

PROBLEM 1 (20 points)

There is a well-known song called 99 Bottles of Beer:

99 bottles of beer on the wall, 99 bottles of beer.
Take one down and pass it around, 98 bottles of beer on the wall.

98 bottles of beer on the wall, 98 bottles of beer.
Take one down and pass it around, 97 bottles of beer on the wall.
…
1 bottle of beer on the wall, 1 bottle of beer.
Take one down and pass it around, no more bottles of beer on the wall.

No more bottles of beer on the wall, no more bottles of beer.
Go to the store and buy some more, 99 bottles of beer on the wall.

Write a program to print out the complete lyrics of this song, using functions to handle discrete tasks as much as possible. Submit a listing of your program and the complete lyrics from the output. Make sure that your song is grammatically correct regardless of the number of bottles.

Note: I suggest writing your own program first; then you might find it interesting to visit the site. http://www.99-bottles-of-beer.net/ and take a look at the 11 or so Python implementations. This website has implementations of the song in 1,500 or more programming languages.

PROBLEM 2 (30 points)

Modify the program in Problem 3 so it prints out the numbers as words. Thus, you will have

Ninety-nine bottles of beer on the wall, ninety nine bottles of beer.
Take one down and pass it around, ninety-eight bottles of beer on the wall
….

This version also must be grammatically correct. Make sure that you have the upper case and lower case letters in the correct places. Submit a listing of your program together with the complete lyrics of the song written out in words. A note on English grammar: the standard when stating numbers as words is to always hyphenate two-word numbers between twenty-one and ninety-nine, and no others. For example, 121 is written out as one hundred twenty-one. **We will discuss this problem in class.**

PROBLEM 3 (10 points)

A string is called a ***palindrome*** if it reads the same forward and backward. For example, "able was I ere I saw elba" is a palindrome, but "Hello World" is not a palindrome. Write a function that takes a string as input and returns True if the string is a palindrome and False if it is not a palindrome. Submit a listing of your program together with some sample output. Note: be sure to test your program with input containing both even and odd numbers of characters; for example “abaaba” and “abacaba” are both palindromes (although they are not English).

Note that in English, character case, punctuation and spaces are normally disregarded in considering palindromes. For example, two famous palindromes (or palindromic sentences) are

> A man, a plan, a canal – Panama
> Madam, I'm Adam

This is a more difficult string manipulation problem than the one assigned – your function does not have to return True for these strings. An even more difficult problem is recognition of semordnilap pairs – this might be assigned as homework later in the semester.

PROBLEM 4 (10 points)

When zeros are used as characters, they are significant in the leftmost position; for example, in zip codes (04469) and UMS employee/student ID numbers (0012345). However, when pasted or entered into a spreadsheet or other software the leading zeros are often dropped because the character string is treated as a number. So it is often the case that moving data between applications involves restoring leading zeros.

In other cases, leading characters are added to a string or number. For example, a check printing application might use leading * characters to pad the check amount to prevent people from adding leading digits – so 1,234.56 might be printed as $****1,234.56.

Write a Python function called leftPad that accepts two parameters:

1. Either a number or a string to left pad
2. The pad character
3. The desired length to pad to

Regardless of whether the first parameter is an int or a string, the function returns a string with zeros prepended to the left of the input so that it is the desired length. For example

```
leftPad('12345','0',7)      #  returns '0012345'
leftPad(1234.56,"*",10)     #  returns '***1234.56'
leftPad("1,234.56","*",1)   #  returns '***1,234.56'
```

If the length of the input is equal to or greater than the desired length, just return the input string. If it is empty, return a string of all pad characters.

```
leftPad('12312312345','0',7)     #  returns '12312312345'
leftPad('125','0',3)             #  returns '125'
leftPad('','#',7)                #  returns '#######'
```

Submit your program listing and output demonstrating that it is correctly implemented. This includes the empty string, correct output with both ints and strings; correct output when the input length is equal to or greater than the requested length.

PROBLEM 5 (25 points) Write a Python program that will produce the following 7-bit ASCII Table. For characters 0-31 (the non-printable control characters) we simply output 'ctrl' instead of a character. Character 127 is the DEL character (from teletype days!), and of course 128 and 129 do not exist in 7-bit ASCII (range 0-127).

Notes:

1. Consider the constant difference between the numbers in row
2. The leftPad() function from problem 4 might be handy. This is a functional equivalent of Python's String.rjust() method. See the very bottom of section 7.1.6 (deprecated functions) in https://docs.python.org/2/library/string.html. We will be looking at more sophisticated methods for outputting strings in the future.

```
                                              ASCII TABLE
---+--0--+--1--+--2--+--3--+--4--+--5--+--6--+--7--+--8--+--9--+
  0|ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  1|ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  2|ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |ctrl |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  3|ctrl |ctrl |     |  !  |  "  |  #  |  $  |  %  |  &  |  '  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  4|  (  |  )  |  *  |  +  |  ,  |  -  |  .  |  /  |  0  |  1  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  5|  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  :  |  ;  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  6|  <  |  =  |  >  |  ?  |  @  |  A  |  B  |  C  |  D  |  E  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  7|  F  |  G  |  H  |  I  |  J  |  K  |  L  |  M  |  N  |  O  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  8|  P  |  Q  |  R  |  S  |  T  |  U  |  V  |  W  |  X  |  Y  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  9|  Z  |  [  |  \  |  ]  |  ^  |  _  |  `  |  a  |  b  |  c  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
 10|  d  |  e  |  f  |  g  |  h  |  i  |  j  |  k  |  l  |  m  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
 11|  n  |  o  |  p  |  q  |  r  |  s  |  t  |  u  |  v  |  w  |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
 12|  x  |  y  |  z  |  {  |  |  |  }  |  ~  | DEL | n/a | n/a |
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Please submit a listing and sample run of your program. When you put the table into a document be sure to use a monospaced font such as Courier New so that the table is displayed correctly.

Hints: 1. build an entire line in a string variable rather than using print statements for each part.
2. While you could use loops to build the separator line and similar components, it is also fair game to copy them from this document and paste them into Python code to define a string variable (e.g., `separatorLine = '…'`)

PROBLEM N (5 points) HOMEWORK ASSISTANCE STATEMENT

- List the names of people who helped you, and explain what sort of help they provided.
- List any resources that you used (e.g., websites, books) other than the textbooks and lecture notes.