CSL 101 - Practice Questions for Quiz 1 and Minor 1

January 25, 2013

1. A sequence of digits is called a *Palindrome* if it reads same from the start and from the end. Some examples of palindrome are 808, 111, 5665 etc.

   (a) Given an input number n, write a recursive Python program to check if n is palindrome or not.

   (b) Given an input number m, use the function written above to find the smallest number $n \geq m$ which is a palindrome.

   Note: You are not allowed to use any string operations for this problem.
   Hint: You may need to write more than one function for part (a) above. Think about how you will extract the most significant and least significant digits of a number.

2. Write a recursive function in Python to find the sum of first n positive integers.

3. Consider Euclid's algorithm to calculate the greatest common divisor (gcd) of two numbers a and b by recursively expressing $gcd(a, b)$ in terms of $gcd(b, a \ mod \ b)$ until the base condition is reached.

   (a) Write a recursive program which implements the above algorithm in Python.

   (b) Use PMI to prove the correctness of your program in the part above. You can use the fact that $gcd(a, b) = gcd(b, a \ mod \ b)$ when $0 < b \leq a$. Remember to clearly mention the variable on which you are going to use PMI.

4. A list can be thought of as a type of data element (structure) which stores a sequence of numbers. For instance, $L = [4, 5, 3, 10, 7]$ is a list with 5 elements in it. A list is said to be sorted if the elements in it are arranged in the increasing order. For instance, the list L above is not sorted but list $L' = [3, 4, 5, 7, 10]$ is a sorted list. Assume that you are given the following operations on a list data structure:

   (a) **size(list):** Takes as input a list. Returns the number of elements in the list. E.g., $size(L)$ will return 5.

   (b) **subseq(list,i,j):** Takes as input a list and two indices. Returns a subsequence of the list starting at index i and ending at index j. E.g., $subseq(L, 2, 4)$ will return a list $L_s = [5, 3, 10]$. The function returns an empty list if any of the indices i or j are out of range or if $i > j$.

   (c) **merge(list1,list2):** Takes as input two sorted lists. Returns a sorted list by merging the elements (in the correct order) in the two lists. For instance, if $L1 = [2, 5, 8]$ and $L2 = [4, 6, 7]$ then $merge(L1, L2)$ will return $L3 = [2, 4, 5, 6, 7, 8]$. The answer is not well defined if the input lists to the $merge$ function are not sorted.

(a) Using above constructs, write a recursive function to sort a list. The basic idea is to divide the list into two halves, sort each half recursively and then merge the resulting lists.

(b) The time requirement of the above function (when implemented correctly) follows the following recurrence relationship:

$$T(n) = K * n + 2 * T(n/2)$$

where n is the size of the initial list and K is some constant. Using this equation, calculate the time complexity of the program using the order of the growth notation.

5. The value of $sin(x)$ can be approximated using the following expression around $x = 0$:

$$sin(x) = x - x^3/3! + x^5/5! + \cdots + (-1)^{k+1} * x^{2k-1}/(2k-1)! + \cdots$$

Write a recursive Python program which takes inputs n and v from the user and evaluates the n^{th} term of this expression at $x = v$.

6. Find the fallacy in the following proof by PMI.
Theorem: Given any collection of n apples, If at least one of them is bad, then all n of them are bad.
Proof: The statement is obviously true for $n = 1$. The step from k to $k + 1$ can be illustrated by going from $n = 3$ to $n = 4$. Assume, therefore, that the statement is true for $n = 3$ and let $A1$, $A2$, $A3$, $A4$ be four apples, at least one of which, say $A1$, is bad. Taking $A1$, $A2$ and $A3$ together and using the fact that the statement is true when $n = 3$, we find that $A2$ and $A3$ are also bad. Repeating the process with $A1$, $A2$ and $A4$, we find that $A4$ is bad. Thus all four apples in the group are bad. A similar argument allows us to make the step from k to $k + 1$ in general.

7. Place value system is a beautiful way to represent numbers using a finite number of symbols. There exist other systems such as the Roman numeral system which are not based on this idea and hence, can be quite cumbersome. For instance, think of representing 121 using the Roman numeral system. What about 12321? What about 1234321? And 123454321? You get the idea. Read this wikipedia article to know more about the place value system:
`http://en.wikipedia.org/wiki/Place-value_notation`
Though base 10 is almost ubiquitous for representing numbers using the place value system, we can use any base $b \geq 2$ to represent numbers in a similar way. For instance, the representation inside a computer is base 2 (i.e. a sequence of 0's and 1's). In the following program, we will examine how to convert numbers from one base to another.

Write a program which takes as input a base $b1$ ($2 \leq b1 \leq 10$), a number n ($n \geq 0$) represented in base $b1$, and a target base $b2$ ($2 \leq b2 \leq 10$). It then converts the number n to a representation in base $b2$. Make sure to flag an error if n is not a valid representation in base $b1$.

Note: You may want to use the following operations on strings to format the output.

(a) '+' operator takes two input strings and returns the concatenated string. Consider the following piece of code in Python:

```
w1 = 'Hello_'
w2 = 'World'
w3 = w1 + w2
print w3
```

```
--------------------
output => Hello_World
----------------------
```

(b) str(<int>) is a built in function which takes an integer as argument and returns its string representation. Consider the following piece of code in Python:

```
d = 5
word = str(d)
sentence = 'The digit is ' + word
print sentence

--------------------
output => The digit is 5
----------------------
```