CSL 210

Assignment no 1 (Recursion)

(Total marks 20)

Deadline: 31st Jan 2021 11.59 pm

Submission details: Submit to the following email id: **marassignments@gmail.com**
With the Subject Line: OCDS A1 RollNo (e.g. OCDS A1 BT18EEE010)

Note: Each question carries different marks. So, note the marks allotted to each question.

1. Count the number of consonants in a string in a recursive manner. (1 mark)
2. Find the first capital letter in a string using recursion (1 mark)
3. Write a program to simulate the strcmp function using recursion. Use it to sort two strings in lexicographic order. (Lexicographical order means alphabetically).

i/p:	o/p:
acd	aaa
aaa	abc
mnp	acd
mno	mno
abc	mnp

(3 marks)

4. The digital root of an integer n is defined as the result of summing the digits repeatedly until only a single digit remains. For example, the digital root of 1729 can be calculated using the following steps:
   Step 1: 1+7+2+9 = 19
   Step 2: 1+9 = 10
   Step 3: 1+0 = 1
   Because the total at the end of step 3 is the single digit 1, that value is the digital root. Write a function digitalRoot(n) that returns the digital root of its argument. Write the function recursively without using any explicit loop constructs.
   (2 marks)
5. Print the following series using recursion
   1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1
   (1 mark)
6. This question involves a game with teddy bears. The game starts when I give you some bears. You can then give back some bears, but you must follow these rules (where n is the number of bears that you have):
   a. If n is even, then you may give back exactly n/2 bears. (To test whether n is even, use the expression ((n % 2) == 0)).
   b. If n is divisible by 3 or 4, then you may multiply the last two digits of n and give back this many bears. (By the way, the last digit of n is n%10, and the next-to-last digit is ((n%100)/10),
   c. If n is divisible by 5, then you may give back exactly 42 bears.

   The goal of the game is to end up with EXACTLY 42 bears.
   For example, suppose that you start with 250 bears. Then you could make these moves:
   --Start with 250 bears.
   --Since 250 is divisible by 5, you may return 42 of the bears, leaving you with 208 bears.

--Since 208 is even, you may return half of the bears, leaving you with 104 bears.
--Since 104 is even, you may return half of the bears, leaving you with 52 bears.
--Since 52 is divisible by 4, you may multiply the last two digits (resulting in 10) and return these 10 bears. This leaves you with 42 bears.
--You have reached the goal!
Write a recursive method to meet this specification:

Postcondition: A true return value means that it is possible to win the bear game by starting with n bears. A false return value means that it is not possible to win the bear game by starting with n bears.

Examples:
bear(250) is true (as shown above),
bear(42) is true,
bear(84) is true ,
bear(53) is false
bear(41) is false

Hint: THINK RECURSIVELY (5 marks)

7. Given an array of 10 numbers, sort the elements at even indices in ascending order. Sort the elements at odd indices using descending order. Note that the index starts from zero.

e.g i/p: 56, 34, 12, 98, 5, 132, 44, 19, 33, 55

o/p: 5, 132, 12, 98, 33, 55, 44, 34, 56, 19

(Indicate which algorithm/algorithms you are using.)

(2 marks)

8. Count the number of inversions in an array in n log n steps. *Inversion Count* for an array indicates – how far (or close) the array is from being sorted in a particular manner say ascending. If array is already sorted then inversion count is 0. If array is sorted in reverse order that inversion count is the maximum.
Formally speaking, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j
Example:
The sequence 2, 4, 1, 3, 5 has three inversions (2, 1), (4, 1), (4, 3).

(5 marks)