Introduction to Programming and Data Structures in Python

Duration: 2 Hours
Maximum Marks: 30

1. Describe a python function `braid` which takes three lists `I,F` and `S` as input and outputs a single list. The list `I` is a sequence of positive integers. It merges the lists `F` and `S` as follows: the first `I[0]` entries of the answer are from `F`, the next `I[1]` entries are from `S`, the next `I[2]` entries are from `F` and so on. Here are some examples:

```
braid([1,0,1,2],['a','b','c'],[13,14,15,16] = ['a','b',13,14]
braid([1,2,3,2],['a','b','c'],[13,14,15,16] = ['a',13,14]
braid([1,2,2,3],['a','b','c'],[13,14,15,16] = ['a',13,14,'b','c']
braid([1,0,2,2],['a','b','c'],[13,14,15,16] = ['a','b','c',13,14]
braid([],['a','b','c'],[13,14,15,16] = []
```

(5 marks)

2. Program a variant of the binary search function, `intsearch`, whose behaviour is as follows: Given a sorted list $l = [a_0, ..., a_n]$ and a value v, it returns a pair (l, r) where l is the largest index such that $a_l < v$ (if there no index then l is -1) and r is the smallest index such that $a_r > v$ (if there is no such index then r is $n + 1$). Your algorithm should run in time logarithmic in the length of the list.

```
intsearch([],3) = (-1,0)
intsearch([1],3) = (0,1)
intsearch([3],3) = (-1,1)
intsearch([1,2,2,2,3,3,4],2) = (0,4)
```

(5 marks)

3. A *majority element* in the array is an element that appears in at least $N/2$ positions. You are given a sorted array and your task is to find if it contains a majority element and identify it if it does. Your algorithm should run in time $O(logN)$. *(3 marks)*

4. Write a python program, using the regular expressions library, that does the following: It takes the name of a file as a command-line argument. It reads the file and outputs a modified version of the text contained in the file in which the first letter of each sentence is capitalized. Suppose the file `input.txt` contained the following:

```
This is a sample text. some sentences begin with lower-case letters so
that you have something to do. The text may contain numbers such as 12.40 but
as is evident that will cause no ambiguity. But if it contains python
code such as self.getmyname() then you should be careful not to
capitalize the g.

so, look for a "." followed by some spaces or ...
```

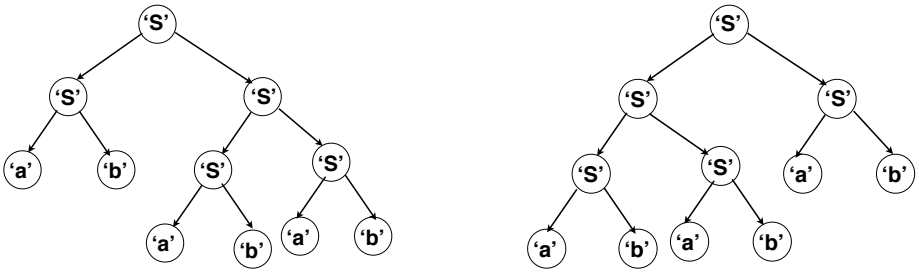
Then the output of your program when given `input.txt` as argument should be:

```
This is a sample text. Some sentences begin with lower-case letters so
that you have something to do. The text may contain numbers such as 12.40 but
as is evident that will cause no ambiguity. But if it contains python
code such as self.getmyname() then you should be careful not to
capitalize the g.

So, look for a "." followed by some spaces or ...
```

(6 marks)

5. Write a python function `levels` that takes a binary tree as input and returns a list of lists, giving the elements of the tree in each level as a list. Here are some examples to illustrate this:



For the tree on the left the answer is

`[['S'],['S','S'],['a','b','S','S'],['a','b','a','b']]`.

and for the tree on the right the answer is

`[['S'],['S','S'],['S','S','a','b'],['a','b','a','b']]`. *(5 marks)*

6. You have file `batch.csv` which lists the loginid and the year in which the each student joined CMI. For each year there is a file `<year>.csv` listing the loginid of each student who joined the Institute in that year along with the year that student graduated.

   Your aim is write a python function `overlapped` which takes a loginid (as a string) as argument and returns a list containing the loginids of all the students whose stay at CMI overlapped with the stay of the given student (i.e. they were at CMI together for at least one year). *(6 marks)*