

# Assignment 8a - Recursion

1. Take as input str, a string. We are concerned with all the possible subsequences of str. E.g. "abcd" has following subsequences "", "d", "c", "cd", "b", "bd", "bc", "bcd", "a", "ad", "ac", "acd", "ab", "abd", "abc", "abcd".
  - a. Write a recursive function which returns the count of subsequences for a given string. Print the value returned.
  - b. Write a recursive function which returns an ArrayList of subsequences for a given string. Print the value returned.
  - c. Write a recursive function which prints all possible subsequences for a given string (void is the return type for function).
2. Take as input str, a string. We are concerned with all the possible ascii-subsequences of str. E.g. "ab" has following ascii-subsequences "", "b", "98", "a", "ab", "a98", "97", "97b", "9798".
  - a. Write a recursive function which returns the count of ascii-subsequences for a given string. Print the value returned.
  - b. Write a recursive function which returns an ArrayList of ascii-subsequences for a given string. Print the value returned.
  - c. Write a recursive function which prints all possible ascii-subsequences for a given string (void is the return type for function).
3. Take as input str, a string. str represents keys pressed on a nokia phone keypad. We are concerned with all possible words that can be written with the pressed keys. E.g. "12" can produce following words "ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf".
  - a. Write a recursive function which returns the count of words for a given keypad string. Print the value returned.
  - b. Write a recursive function which returns an ArrayList of words for a given keypad string. Print the value returned.
  - c. Write a recursive function which prints all possible words for a given keypad string (void is the return type for function).
4. Take as input str, a string. We are concerned with all possible permutations of characters in str. E.g. "abc" can produce following words "abc", "acb", "bac", "bca", "cab", "cba".
  - a. Write a recursive function which returns the count of different permutations for a given string. Print the value returned.
  - b. Write a recursive function which returns an ArrayList of permutations for a given string. Print the value returned.
  - c. Write a recursive function which prints all possible permutations for a given string (void is the return type for function).
5. Google Tower of Hanoi.
  - a. Write a recursive function which returns number of steps required to solve the tower of Hanoi problem for N discs.



# Assignment 8a - Recursion

---

- b. Write a recursive function which prints the steps required to solve the tower of Hanoi problem for N discs.
6. Take as input N, a number. Take N more inputs and store that in an array.
  - a. Write a recursive function which counts the number of ways the array could be split in two groups, so that the sum of items in both groups is equal. Each number in the array must belong to one of the two groups. Print the value returned.
  - b. Write a recursive function which keeps track of ways an array could be split in two groups, so that the sum of items in both groups is equal. Each number in the array must belong to one of the two groups. Return type of function should be `ArrayList<String>`. Print the value returned.  
E.g. for `[1, 3, 5, 7, 0]` the returned `ArrayList` will contain `"[1, 7, 0] and [3, 5]"` and `"[1, 7] and [3, 5, 0]"` as its constituents.
  - c. Write a recursive function which keeps track of ways an array could be split in two groups, so that the sum of items in both groups is equal. Each number in the array must belong to one of the two groups. Return type of function should be `void`. Print the two groups, each time you find a successful split.
7. Take as input N, a number. Take N more inputs and store that in an array. Take as input target, a number
  - a. Write a recursive function which counts the number of subsets of the array which sum to target. Print the value returned.
  - b. Write a recursive function which returns the subsets of the array which sum to target. Return type of function should be `ArrayList<ArrayList<Integer>`. Print the value returned.
  - c. Write a recursive function which prints subsets of the array which sum to target. Return type of functions is `void`.
8. Take as input N, a number. Write a recursive function which prints counting from 1 to N in lexicographical order. In lexicographical (dictionary) order 10, 100 and 109 will be printed before 11.
9. Take as input str, a string. Write a recursive function which prints all the words possible by rearranging the characters of this string which are in dictionary order larger than the given string.
10. Take as input str, a string. Write a recursive function which returns all the words possible by rearranging the characters of this string which are in dictionary order smaller than the given string.

