1. A child is running up a staircase with n steps and can hop either 1 step, 2 steps, or 3 steps at a time. Implement a method to count how many possible ways the child can run up the stairs.

****

2. Imagine a robot sitting on the upper left corner of grid with r rows and c columns. The robot can only move in two directions, right and down, but certain cells are "off limits" such that the robot cannot step on them. Design an algorithm to find a path for the robot from the top left to the bottom right.

****

3. Write a method to return all subsets of a set.

****

4. Write a recursive function to multiply two positive integers without using the *operator.You can use addition, subtraction, and bit shifting, but you should minimize the number of those operations

****

5. In the classic problem of the Towers of Hanoi, you have 3 towers and N disks of different sizes which can slide onto any tower. The puzzle starts with disks sorted in ascending order of size from top to bottom (i.e., each disk sits on top of an even larger one). You have the following constraints:

   (1) Only one disk can be moved at a time.

   (2) A disk is slid off the top of one tower onto another tower.

   (3) A disk cannot be placed on top of a smaller disk. Write a program to move the disks from the first tower to the last using stacks.

****

6. Write a method to compute all permutations of a string whose characters are not necessarily unique. The list of permutations should not have duplicates.

****

7. Write an algorithm to print all ways of arranging eight queens on an 8x8 chess board so that none of them share the same row, column, or diagonal. In this case, "diagonal" means all diagonals, not just the two that bisect the board.

****

8. You have a stack of n boxes, with widths wi , heights hi , and depths di . The boxes cannot be rotated and can only be stacked on top of one another if each box in the stack is strictly larger than the box above it in width, height, and depth. Implement a method to compute the height of the tallest possible stack. The height of a stack is the sum of the heights of each box.

****

9. Given a boolean expression consisting of the symbols 0 (false), 1 (true), & (AND), I (OR), and /\ (XOR), and a desired boolean result value result, implement a function to count the number of ways of parenthesizing the expression such that it evaluates to result.

EXAMPLE

countEval("l /\01011", false) -> 2

countEval("0&0&0&1/\ll0", true) -> 10

****

10. You are given two sorted arrays, A and B, where A has a large enough buffer at the end to hold B. Write a method to merge B into A in sorted order.

****

11. Write a method to sort an array of strings so that all the anagrams are next to each other.

****

12. Given a sorted array of n integers that has been rotated an unknown number of times, write code to find an element in the array. You may assume that the array was originally sorted in increasing order.

lnput:findSin{lS, 16, 19, 20, 25, 1, 3, 4, 5, 7, 10, 14}

Output: 8 (the index of 5 in the array)

****

13. Given a sorted array of strings that is interspersed with empty strings, write a method to find the location of a given string.

Input: ball,{"at","" , '"' , "" , "ball", "" , "" , "car", "" , '"' , "dad", '"' ,""}

Output:4

****

14. Imagine you have a 20 GB file with one string per line. Explain how you would sort the file.

****

15. Given an input file with four billion non-negative integers, provide an algorithm to generate an integer that is not contained in the file. Assume you have 1 GB of memory available for this task. FOLLOW UP What if you have only 10 MB of memory? Assume that all the values are distinct and we now have no more than one billion non-negative integers.

****

16. You have an array with all the numbers from 1 to N, where N is at most 32,000. The array may have duplicate entries and you do not know what N is. With only 4 kilobytes of memory available, how would you print all duplicate elements in the array?

****

17. Given an M x N matrix in which each row and each column is sorted in ascending order, write a method to find an element.

18.Imagine you are reading in a stream of integers. Periodically, you wish to be able to look up the rank of a numberx (the number of values less than or equal to x). lmplementthe data structures and algorithms to support these operations. That is, implement the method track (int x), which is called when each number is generated, and the method getRankOfNumber(int x), which returns the number of values less than or equal to x (not including x itself).

EXAMPLE

Stream (in order of appearance): 5, 1, 4, 4, 5, 9, 7, 13, 3

getRankOfNumber(l)= 0

getRankOfNumber(3) = 1

getRankOfNumber(4) = 3

****

19. In an array of integers, a "peak" is an element which is greater than or equal to the adjacent integers and a "valley" is an element which is less than or equal to the adjacent integers. For example, in the array {5, 8, 6, 2, 3, 4, 6}, {8, 6} are peaks and {5, 2} are valleys. Given an array of integers, sort the array into an alternating sequence of peaks and valleys.

Input: {5, 3, 1, 2, 3} Output: {5, 1, 3, 2, 3}

****